NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

WEB-BASED DISSEMINATION SYSTEM FOR THE TRUSTED COMPUTING EXEMPLAR PROJECT

by

Douglas Robert Kane Jr.

June 2005

Thesis Advisor:	Cynthia E. Irvine
Co-Advisor:	Thuy D. Nguyen

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** June 2005	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE: Web-Based Dissemination System for the Trusted Computing Exemplar Project			**5. FUNDING NUMBERS**
6. AUTHOR(S) Douglas R. Kane Jr.			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			**12b. DISTRIBUTION CODE**

13. ABSTRACT (maximum 200 words)

Open dissemination of the Trusted Computing Exemplar (TCX) project is needed. This dissemination must include methods to provide secure web access to project material, integrity verification of data, and group-based access controls. Because previously developed dissemination systems do not meet these requirements, a hybrid web-based dissemination system is necessary.

The development of the TCX Dissemination System requirements involved the analysis of assumptions, threats, policies, and security objectives for the system and its environment based on the Common Criteria methodology. The requirements yielded a design specification that included a dissemination application that uses XML capabilities for redaction and preparation of releasable materials. This led to the creation of an initial implementation to satisfy a subset of the TCX dissemination requirements. Future work was identified for a subsequent implementation that fulfills additional project requirements.

The complete implementation of the dissemination environment described in this thesis will provide a seamless dissemination interface for the TCX project. The Dissemination System provides an example of how controlled information can be organized and made available on the web. When combined with TCX project results, it supports the assured information sharing objectives of the Department of Defense Global Information Grid vision.

14. SUBJECT TERMS Trusted Computing Exemplar, web-based dissemination system, Common Criteria, secure delivery, XML-based access control, document redaction			**15. NUMBER OF PAGES** 152 **16. PRICE CODE**
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited.

WEB-BASED DISSEMINATION SYSTEM FOR THE TRUSTED COMPUTING EXEMLAR PROJECT

Douglas Robert Kane Jr.
Ensign, United States Navy
B.S., United States Naval Academy, 2004

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL

June 2005

Author: Douglas Robert Kane Jr.

Approved by: Cynthia E. Irvine
Thesis Advisor

Thuy D. Nguyen
Co-Advisor

Peter J. Denning
Chairman, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Open dissemination of the Trusted Computing Exemplar (TCX) project is needed. This dissemination must include methods to provide secure web access to project material, integrity verification of data, and group-based access controls. Because previously developed dissemination systems do not meet these requirements, a hybrid web-based dissemination system is necessary.

The development of the TCX Dissemination System requirements involved the analysis of assumptions, threats, policies, and security objectives for the system and its environment based on the Common Criteria methodology. The requirements yielded a design specification that included a dissemination application that uses XML capabilities for redaction and preparation of releasable materials. This led to the creation of an initial implementation to satisfy a subset of the TCX dissemination requirements. Future work was identified for a subsequent implementation that fulfills additional project requirements.

The complete implementation of the dissemination environment described in this thesis will provide a seamless dissemination interface for the TCX project. The Dissemination System provides an example of how controlled information can be organized and made available on the web. When combined with TCX project results, it supports the assured information sharing objectives of the Department of Defense Global Information Grid vision.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

ABBREVIATIONS AND ACRONYMS

ASCII	American Standard Code for Information Interchange
CA	Certificate Authority
CC	Common Criteria
CI	Configuration Item
CISR	Center for Information Systems Security Studies and Research
CM	Configuration Management
CVS	Concurrent Versions System
DA	Dissemination Application
DIE	Documentation Integration Environment
DDF	Document Descriptor File
DS	Dissemination System
EAL	Evaluation Assurance Level
FTP	File Transfer Protocol
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol with SSL
IP	Internet Protocol
IT	Information Technology
NIST	National Institute of Standards and Technology
NPS	Naval Postgraduate School
NSA	National Security Agency
PEM	Privacy Enhanced Mail
PGP	Pretty Good Privacy
RA	Releasing Agent
RIL	Releasable Items List
SKPP	Separation Kernel Protection Profile
SSL	Secure Sockets Layer
TCB	Trusted Computing Base
TCSEC	Trusted Computer Systems Evaluation Criteria

TCX	Trusted Computing Exemplar
TOE	Target of Evaluation
TSF	Target of Evaluation Security Functions
XML	Extensible Markup Language
XSL	Extensible Stylesheet Language

ACKNOWLEDGMENTS

I would like to thank my thesis advisors, Dr. Cynthia Irvine and Thuy Nguyen for all of their help and support on this project. Without their guidance and insight I would not have been able to complete this project. I would like to thank Dr. Cynthia Irvine for her meticulous attention to detail, without which I could not have created a project of such high quality. I would like to thank Thuy Nguyen for helping me to develop my first design specification and implementation. Her knowledge provided an amazing learning experience throughout the thesis process. I would like to thank John Clark and the rest of the TCX project for providing me with the need to create an XML-based dissemination system.

Finally, I would like to thank my friends and family who supported me throughout the entire thesis process.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A primary objective of the Trusted Computing Exemplar (TCX) project is to provide an example of high assurance development that will be available worldwide. In addition, geographically distributed project collaborators and evaluators need access to releasable project artifacts. Thus, a means of online dissemination is required. While other open source projects make available their project material on unrestricted websites, the TCX project requires a more controlled dissemination environment. Specifically, all TCX project materials will have access control markings and their dissemination will be regulated based on those markings. The dissemination system used by the TCX project will need to provide strictly controlled web distributions with configurable access control to different portions of the project results.

A. OVERVIEW OF TCX PROJECT

The goal of the TCX project is to provide an openly distributed worked example of how high assurance trusted computing components can be built [1]. The majority of developers today are not focusing on creating high assurance products, mainly due to lack of training and the early-to-market push. The TCX project was created to show future developers how to apply high assurance development methodology to the construction of products that can be evaluated against the highest evaluation assurance level defined by the Common Criteria [2].

High assurance trusted computing addresses the problems of frontal attacks and subversion. It encompasses the science and engineering required to specify, design, implement, and maintain components which have a high level of confidence against both frontal attacks and system subversion [1]. Systems must meet not only sound security criteria, but also be built in such a way that it is possible to verify the protection mechanisms provided.

The TCX project includes four related activities to spread the knowledge of high assurance development [3]. These four activities are:

- Creation of a reusable high assurance development framework
- Development of a reference-implementation trusted network component

- Support for evaluation of the reference component against the highest assurance criteria as defined in the Common Criteria (EAL7)
- Open dissemination of the results of the first three activities

The reusable development framework and open dissemination objectives establish a set of conditions that the design of the TCX dissemination system must address.

1. Reusable Development Framework

The development framework is made up of a high assurance life cycle framework and a high assurance rapid development environment. The life cycle framework utilized is the spiral life cycle model with additional high assurance properties required by the EAL7 life cycle requirements [3]. The additional life cycle requirements include rigorous configuration management and strict developmental security safeguards [3]. The rapid development environment consists of a documentation integration environment (DIE) used to construct and manage the TCX project documents; development tools and procedures for construction of TCX software; and verification tools and procedures with which to determine with high assurance whether the system that is built is as was defined [3]. The DIE specifies the use of XML for authoring the project documents. XML allows greater document control and paves the way for fine grained access control during dissemination.

2. Open Dissemination

The open dissemination requirements for the TCX project include mechanisms for continuous contribution, evaluation and distribution of various project configuration items and deliverables [1]. Currently the TCX project has no mechanisms in place to fulfill these requirements or to openly distribute project outputs.

B. OVERVIEW OF TCX DISSEMINATION SYSTEM REQUIREMENTS

The TCX Dissemination System is required to disseminate project material over the Internet. To satisfy the open dissemination requirement, the Dissemination System must be web-based. The Dissemination System must be able to disseminate XML documents, source code, and various other file formats with integrity and confidentiality protection. Additionally, the Dissemination System must provide group-based access control for different areas of the online site. Common Criteria high assurance trusted delivery requirements must also be addressed by the TCX Dissemination System.

C. PURPOSE OF STUDY

The purpose of study is to determine the best way to build a dissemination system that satisfies the TCX dissemination requirements. First an analysis of existing dissemination systems is conducted. Based on the results of this analysis, a design for the initial implementation of an XML-centric web-based dissemination system is proposed. This design will include a threat analysis of the proposed system, development of the system requirements, and generation of a high level design specification. The design specification is used to implement an initial prototype. The result of this thesis is a useable dissemination prototype for the TCX project that mostly satisfies the TCX dissemination requirements. The lessons learned from this effort provide crucial insight for future refinement of the prototype.

D. ORGANIZATION OF PAPER

This thesis contains eight chapters and one appendix. Chapter I describes the need for the design and implementation of the TCX Dissemination System. Chapter II provides background material on existing dissemination systems, Common Criteria requirements, Apache web server, and XML tags. Chapter III contains the requirements analysis including the high level design and concept of operation for the Dissemination System and Application. Additionally, it contains the assumptions, policies, threats and security objectives for the Dissemination System and its environment. Chapter IV provides a detailed description of the system requirements and the rationale mapping of the assumptions, threats, policies, and security objectives that created them. Chapter V contains an overview of the system design and a detailed high level functional design specification. Chapter VI describes the initial prototype design to implement the Dissemination System. Finally, Chapter VII contains future work and conclusions. Appendix A contains configuration files from the initial implementation while Appendix B contains selective screen captures from the testing.

THIS PAGE INTENTIONALLY LEFT BLANK

II. BACKGROUND

The background information contained herein is described in four separate sections. The first section contains a survey of existing dissemination systems to determine whether a new dissemination system must be created or if an existing one would be suitable. The next section addresses the Common Criteria trusted delivery requirements. This section first reviews the trusted distribution requirements from the TCSEC and then the trusted delivery requirements from the Separation Kernel Protection Profile, which is based on the Common Criteria. The third section discusses the usefulness of the Apache Web Server and its security services. The last section provides an overview of XML and some of its potential uses within the TCX project.

A. SURVEY OF EXISTING DISSEMINATION SYSTEMS

1. Introduction

Before developing the Dissemination System for the TCX project, a survey of existing online dissemination systems was conducted. This was done for two reasons: first, if a system exists that fulfills the needs for the TCX project dissemination, then there is no need to create a new one, and second, other dissemination systems might provide guidance and lessons learned that might be applied to the development of the TCX Dissemination System.

To guide the survey, specific requirements for the TCX project must be identified. First, the Dissemination System must be able to support multiple user groups because the developers realized that certain information should only be provided to specific users. Second, since the project will be evaluated for an EAL7 certification, evaluation and validation evidence must be available. Furthermore, most documentation for the TCX project is written in XML. The Dissemination System must therefore be able to distribute XML documents, in addition to the TCX kernel software, and other non-XML material. These requirements were all taken into account while analyzing other online dissemination systems. This survey examined dissemination techniques implemented by university projects and selected open source projects available online. Specifically, documentation availability, security methods, user access and data download mechanisms were examined.

2. EROS-OS

The first university project examined was the Extremely Reliable Operating System or EROS-OS. All project material examined can be found online and the information here came from the EROS-OS website [4]. The project was originally implemented at the University of Pennsylvania, but was later moved to Johns Hopkins University. The primary goal of this project is to create a small, secure, real-time operating system. The documentation available for this project was limited, and the background information was not relevant to this study. Warning banners displayed that documents might not reflect the actual implementation and most data available were draft versions of documents. All documentation available online had no form of integrity verification. The site had no restricted sections, therefore no identification and authentication methods were employed. The EROS-OS team seemed to communicate solely via mailing lists. There are multiple mailing lists and some required that the members of the list be part of the development team.

To obtain the actual project files and the kernel, a user must use the OpenCM repository [4]. OpenCM is a free program that the user must download and install on his or her own system before using. To use OpenCM with the project, the user also has to install the anonymous access key for non-developers. This complex installation of programs would not be efficient for the TCX project.

3. Fiasco Microkernel

The next project examined was the Fiasco Microkernel developed by the Dresden University of Technology. This dissemination system was difficult to navigate but can be found online [5]. There was limited documentation found on the website, and much of it seemed incomplete. These documents were available for download in an Adobe Acrobat format. Again, the project used mailing lists for project discussion, but the mailing lists seemed unrestricted. To receive the project kernel the user must connect through a remote CVS system. Once the tar archive is downloaded, additional modules are needed for use of the project. The tar archive also had an MD5 checksum so the user can check the integrity of the file downloaded. While this project did have additional integrity protection in the form of MD5 checksums, it did not provide the access controls necessary for the TCX project.

4. Apache Web Server

The Apache Web Server is one of the most used web servers on the Internet; and thus it should have an effective method of dissemination. The Apache Web Server is available online and is already prepackaged with many Linux distributions [6]. The website offered extensive documentation in the form of HTML documents without integrity verification. All of the documentation was publicly accessible. There were tar archive downloads available for installing Apache on most operating systems. Also, integrity checks were available for the downloaded tar archive as either PGP or MD5 checksums. The Apache group also had multiple mailing lists for different groups of users of the system. While the Apache project's dissemination scheme could satisfy some of the TCX project dissemination goals, it does not have document integrity which is essential to the TCX project validation process.

5. OpenBSD

OpenBSD is a free UNIX-like operating system available online [7]. The online documentation is available in the form of *man* pages. Additionally, there is a daily change log online that tracks all changes made to the system. This feature provides thorough feedback that can otherwise be hard to track by viewers of the project. Group mailing lists were also accessible online.

The online site offered project material from multiple CVS servers in addition to FTP and CD sets; however, the anonCVS server is stated as the preferred method of downloading the project. When the user accesses the CVS server it updates the local copy of the software with changes made to the current OpenBSD sources. While this is an effective way to assure that the user has the most up-to-date version, it would not be necessary for the TCX project. Furthermore, the user cannot simply download a tar archive of the project from the website, and instead requires the CVS interface to receive all files easily. CVS does offer an encrypted channel for transmission, but there is no mention of integrity checking of the documents available for download.

6. Other Projects and Conclusions

Other surveyed projects were the OpenSSL Project and the Linux Kernel. These projects provided no additional methods compared to the previous projects discussed.

They utilized similar distribution options and mailing lists and neither had any unique methods.

None of the projects surveyed completely fulfilled the initial goals of the TCX Dissemination System. Specifically, the TCX Dissemination System must implement group-based access controls and be capable of processing XML documents. Furthermore, the TCX project does not utilize the CVS interface which was widely used by other projects. A completely new system must be developed to meet the TCX project goals for proper dissemination of project material.

B. COMMON CRITERIA TRUSTED DELIVERY REQUIREMENTS

Since the TCX project will be evaluated against the Common Criteria, it needs to meet certain delivery requirements. These requirements were first specified in the Trusted Computer Systems Evaluation Criteria (TCSEC) in 1985 for the Class A1 Trusted Computing Base (TCB) [8]. The TCSEC was written and maintained by the Department of Defense and was the precursor to the Common Criteria. It provided a metric for evaluating the effectiveness of security controls built into computer systems. The Common Criteria is the newer, internationally accepted metric used today. The TCSEC trusted distribution requirements will be discussed first, followed by the trusted delivery requirements as written in the Separation Kernel Protection Profile (SKPP) [9].

1. Trusted Distribution from the TCSEC

Trusted Distribution as specified by the TCSEC has two parts. First, the TCSEC states that there must be an automatic control system and distribution facility to maintain the integrity between the master data describing the current version of the system and the on-site master copy for the code of the current version [8]. While this wording is not extremely clear it is explained in the *Guide to Understanding Trusted* Delivery: “all of the distribution material must arrive to a customer site exactly as intended by the vendor without any alterations” [10]. This guarantees that the onsite version must match the master copy.

The second part requires that, there must be procedures to assure that all updates distributed to a customer are exactly the same as the master version [8]. While this might seem more straightforward, the word “procedures” is never specified. In some cases this

can be an external signature verification tool or a form of encryption. But whatever the procedures are they must exist to ensure that the master and customer version match.

Trusted Distribution protects the TCB from two main threats. The first threat is someone tampering with the TCB during its movement. The second threat protects against the distribution of a counterfeit system [10]. These two threats exist for any distributed system and thus these high assurance requirements have not changed significantly over the years.

2. Trusted Delivery from SKPP

The Separation Kernel Protection Profile is written to describe the requirements for a class of high assurance kernels. The TCX Separation Kernel is required to conform to the SKPP and thus the SKPP requirements for trusted delivery must be supported by the TCX Dissemination System. In the Common Criteria paradigm, the system that is the subject of evaluation is referred to as a Target of Evaluation (TOE).

The procedures outlined in the SKPP are for both the initial distribution and subsequent updates to the TOE and its components. Similar to the first requirement from the TCSEC, the on-site version of the TOE must match the master distribution version. Additionally, the SKPP requires that the TOE include procedures and/or tools to verify that the on-site version of the TOE matches the master version [9]. Whereas the TCSEC specifies that a tool must be used to verify the integrity of the system, the SKPP explicitly requires the tool to be part of the evaluation. It states, “such a verification tool may be configured to execute on the TOE (but not as part of the TSF) or on other hardware, but in either case the tool and the hardware that it runs on are evaluated as part of the TOE” [9]. The objective of this requirement is to ensure that the TOE and its components must be delivered to the customer environment securely. This results in the user having a complete and verified TOE. If the TOE or any of its components are modified after delivery or if the TOE is incomplete it will not be in an evaluated configuration [9].

The SKPP meets the requirements for trusted delivery by requiring cryptographic signature services and hashing functions to protect the integrity of the TOE when distributing versions of the TOE to the user site. Additionally, independent channels

must be used to deliver both the TOE and the cryptographic keying materials used to verify the TOE distribution [9].

While the TCSEC sets the standards for trusted distribution, the SKPP helps to clarify and give specifics as to the requirements for trusted delivery. The SKPP builds upon the infrastructure created by the TCSEC.

C. APACHE SECURITY SERVICES

1. Why Apache?

The Apache Web Server was chosen to host the TCX Dissemination System website for three reasons: 1) it is free, 2) it is the most used web server on the Internet, and 3) it provides a variety of security services. The security services that provide the robustness of the web server are audit logging, authentication, and module add-ons, particularly OpenSSL [6]. Each of these services will be discussed in the subsequent sections.

2. Audit Logging

Apache provides extensive logging options for the website. The logging options allow easy auditing of both access to the site and errors encountered with website operation. These configurable audit functions provide all foreseeable auditing necessary for the TCX project. The lowest level of audit logging will record every action, from opening and reading the configuration file to user interaction with the site. This level of logging, called the debug level, provides the most detailed logs possible on the Apache system without the implementation of third party logging mechanism. The flexibility of the Apache audit functionality allows support of different audit policies as may be defined by the TCX project management.

3. Authentication

Apache also provides online authentication methods. While the login prompt varies slightly based on the web browser being used, it provides a common window where the user enters his or her user name and password. The window also contains a configurable message from the web server.

Apache configuration files can specify a single user to have access to a website. Additionally, Apache allows groups of users to be configured with group-specific access to web content. The TCX project will take full advantage of this functionality to

implement group-based access controls. Additionally, the Apache server also hashes the user password in the password file. This is a beneficial security feature because the system administrator will never see the user passwords. The Apache configuration places file authentication requirements in the website configuration file. The password file is a separate file that contains the username and the corresponding password hash. The group file contains the group and the users assigned to the group. Apache allows this as the basic configuration in which all information is sent in the clear. While this might seem like a security vulnerability, the OpenSSL module can mitigate this.

4. Add-on Modules and OpenSSL

Apache allows additional modules to be configured for added functionality. One of the most useful modules is the OpenSSL module. SSL is the Secure Sockets Layer protocol and it operates with HTTP as a service called HTTPS. While HTTP operates on port 80, HTTPS operates on port 443. With SSL, a digital certificate signed by a trusted third party can be used by the web server to authenticate itself to the user. Once this authentication is established an encrypted channel is created between the web server and the user. At this point information can be sent in the clear because it travels through the encrypted channel. SSL also provides logging functionality similar to that of the Apache web server. This includes the logging of SSL errors and SSL accesses to the web site.

5. Apache Conclusion

Apache is an excellent choice for disseminating TCX project material because of both the functional and security services it provides. The audit logging will be sufficient to track website access and error logging for both SSL and non-SSL connections. The authentication methods allow for configurable individual or group access to web pages. Add-on modules provide additional functionalities that can be tailored to specific server needs. In particular, SSL provides the security for both site authentication and encrypted channels necessary for secure dissemination.

D. XML TAG OVERVIEW

1. XML Background

The Extensible Markup Language (XML) is a text and data-formatting language that has a tag-based syntax similar to the Hypertext Markup Language (HTML) [11]. An XML tag is delineated in a document by angled brackets. A start and end tag are required

for proper processing with the end tag containing a forward slash. All data between the start and end tag take the properties that the tags specify. Tags can be arbitrarily nested lending a natural tree structure to XML documents. XML prescribes text styles but also defines data types for cross platform communication. XML documents contain only data, so applications that process XML documents must decide how to display the data based on the embedded tags.

2. Why XML?

To provide traceability between specification and implementation, the TCX project requires most TCX documentation to be written in XML. There were a variety of reasons as to why XML was the best choice. First of all, XML provides a standards-compliant data format. XML has a semi-formally defined structure that can be used to express the structure of documents. XML provides easy methods to publish documents online in addition to providing different presentation modes for the same document. Unlike most documents created by word processors, XML offers fine-grained document control allowing each element to be signed rather than solely signing the entire document. While XML can be written to look like a normal text document, it has the added feature in that it can provide meaning to the information with embedded tags. XML documents can also easily be ported to any platform and maintain their well-formed structure. Finally, XML can separate content from presentation. This allows the authors to focus on content rather than project formatting standards. XML provides a variety of features over common word processor programs and thus was chosen to create the TCX documentation.

3. Document Creation

XML has preset tag libraries that create an XML instance language. The TCX project documentation is created using the DocBook instance language. DocBook was chosen because it is, “particularly well suited to books and papers about computer hardware and software” [11]. However, XML will also be used to create access control tags for the documents. These user-defined tags change the instance language of the document because the user defined tags are not recognized by DocBook.

4. XSL Transformations

XSL is a style sheet language for XML that is commonly used to translate XML documents to HTML documents. This will be utilized in the TCX project so that documentation can be displayed online in a common web browser. In addition to using the standard XSL transformations for DocBook, user-defined tags must be specified as to how they will be display in HTML. This additional configuration will allow TCX project material to be displayed exactly as the authors intended.

E. SUMMARY

After the survey of existing dissemination systems was presented, the need for a new TCX-specific dissemination system was established and the requirements for trusted delivery according to the Common Criteria were described. The benefits of the Apache Web Server and XML with respect to the TCX project were also explained. In the next chapter the requirements analysis of the TCX Dissemination System is discussed.

THIS PAGE INTENTIONALLY LEFT BLANK

III. TCX DISSEMINATION REQUIREMENTS ANALYSIS

This chapter contains the high level description of the Dissemination System including its purpose, concept of operation, conceptual architecture, system access, and XML document handling. Using the Common Criteria methodology, the threat analysis for the Dissemination System is discussed in terms of environmental assumptions, anticipated threats, organizational security policies and security objectives. An overview of the system requirements for secure delivery and dissemination of material is presented last.

A. HIGH LEVEL SYSTEM DESCRIPTION

1. Purpose of TCX Dissemination System

Information from the TCX project will be disseminated across the Internet. Dissemination material will include source code, design specifications, evaluation evidence, user guidance, administrative guidance, and binaries. Trusted delivery requirements must be met for both the dissemination of the TCX kernel and the other dissemination material according to the Separation Kernel Protection Profile [9]. The users will have the material via trusted delivery with a guarantee of the integrity of the source and version. Validators require trusted delivery in order to validate the product.

For the other dissemination ideas discussed previously, the distribution of data in a large zipped file or tar archive via email could be a potential delivery mechanism. This Dissemination System must support multiple groups of users with varying access levels to the data. Because of this, a single tar archive distribution of the material would not be suitable for all data. The Dissemination System must be able to distribute both tar archives and individual files to which specific access controls apply. The different users who will have access to the information include: administrators, collaborators, customers, developers, evaluators, NIST/NSA validators, and the general public.

2. Dissemination Environment Concept of Operation

a. Internal Data Flow

Before data can be accessed by users, it must first be installed on the Dissemination System. The flow of information is displayed in Figure 1.

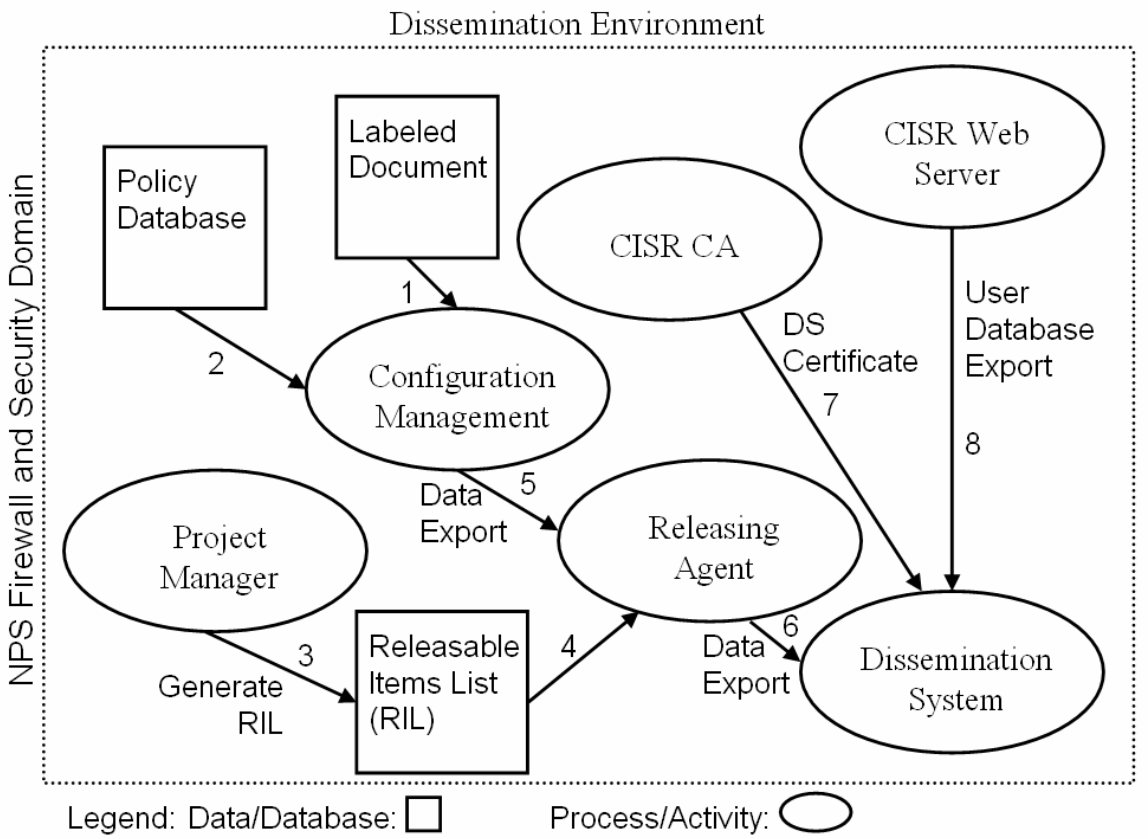


Figure 1. Dissemination Environment Data Flow

First, (step 1) labeled documents or data are submitted to Configuration Management by the developers. There they are entered into the Configuration Management System and signed by the Configuration Manager. Next, (step 2) a dissemination policy database containing release policies for the dissemination material and who can view them is generated by the TCX Configuration Control Board and submitted to Configuration Management. Third, (step 3) the project manager generates a Releasable Items List database (RIL) that itemizes which documents may be released by the Dissemination System. That list is then passed to the Releasing Agent (step 4). The Releasing Agent uses the RIL to request documents from Configuration Management (step 5). In addition, the Releasing Agent obtains the dissemination policy database from Configuration Management (step 5). In step 6 the Releasing Agent exports all releasable material, the RIL, and the policy database to the Dissemination System. After receiving the RIL, the Dissemination Application performs a sweep of the dissemination material repository to assure that all data it controls is releasable in accordance with the current

RIL. If an item is revoked or becomes non-releasable it is moved to a non-releasable items folder on the system. This imposes an additional access control mechanism on the data. The Dissemination System can now perform access control on the releasable material based on the current policy and the access control markings or document descriptor files contained within the data. This generates the webpage repository, through which the user accesses the dissemination material. In step 7 the CISR Certificate Authority exports the signed Dissemination System digital certificate. Finally, in step 8 the user database (described in the next section) is transferred to the Dissemination System. All externally-generated data (steps six through eight) are transferred to the Dissemination System via secure means. At this point, the Dissemination System has all data necessary to properly disseminate project material to appropriate users when requested.

b. User Access Flow

Figure 2 shows the user access flow. Unnumbered steps have been described previously. There are two types of users: public users and registered users. Public users can only access non-proprietary project material. In order to retrieve proprietary project material, users must first register with the CISR Web Server to obtain a unique user name and password before accessing data (step 1). Once registration is complete and the user is authorized as a valid user of the system, the user database will be updated and exported to the Dissemination System (step 2). Additionally, any user groups the user belongs to will be assigned prior to export based on specific information about the user. Users will not be able to gain access to the Dissemination System through the CISR Web Server. The CISR Web Server will also provide a feedback mechanism for the users of the system. The feedback will be processed externally from the Dissemination System within the Dissemination Environment.

Finally, step 3 allows the user to access data on the Dissemination System. The Dissemination Application will produce views for users to download the data by processing the policy and releasable items list. These views will not be kept under the Configuration Management system.

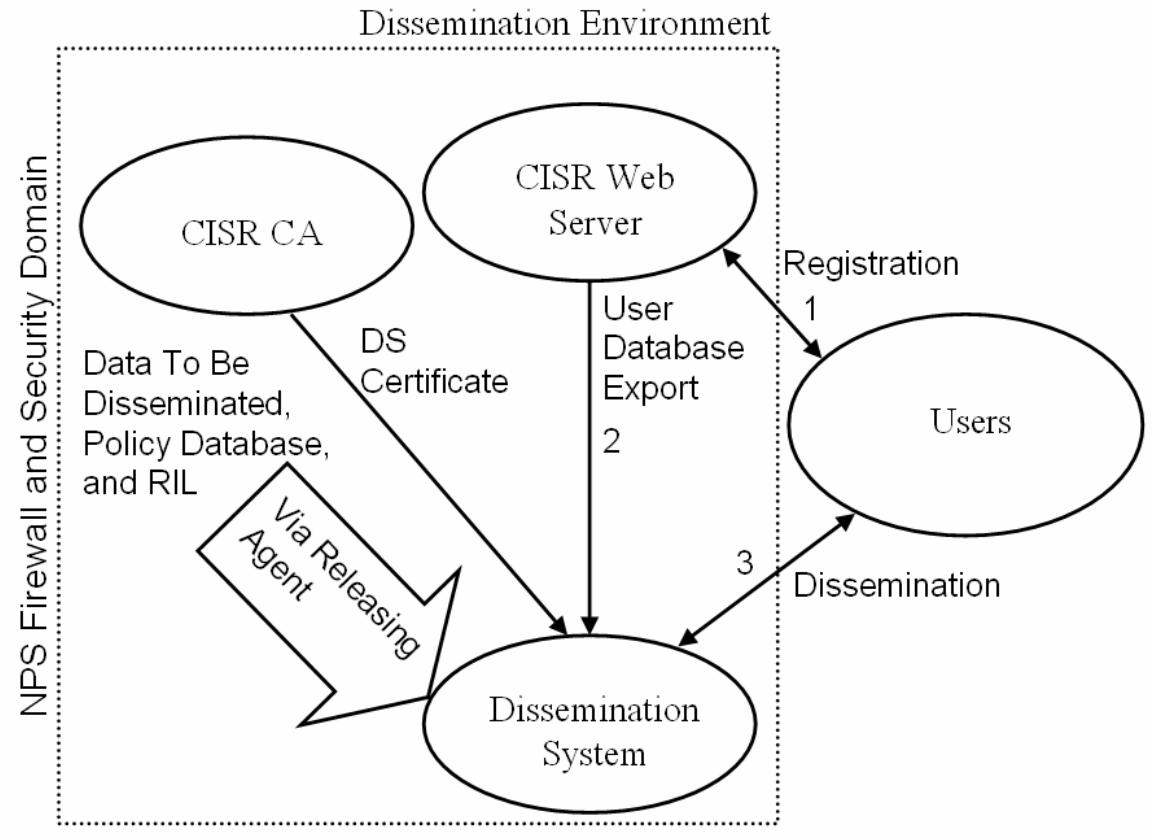


Figure 2. User Data Access on Dissemination System

3. Dissemination System Conceptual Architecture

The Dissemination System is comprised of the operating system, a web server, the Dissemination Application and supporting tools. When data has been approved for release, it will be manually exported from TCX Configuration Management to the Releasing Agent. Then the Releasing Agent will provide the data via secure means to the Dissemination System Administrator who will import it onto the Dissemination System.

The Dissemination Application (DA), located on the Dissemination System, will maintain the confidentiality and integrity of the dissemination data. The DA is responsible for webpage management on the system. Access control markings on documents will be used by the DA to generate the webpage repository. Users will be able to download or view dissemination material through a standard web browser. This two-way connection to the Dissemination System will be the sole external access to dissemination material.

The Dissemination Application will process multiple databases in order to perform its task of properly disseminating data. First, it will have the TCX project dissemination material. Additionally, it will have control of the Releasable Items List and the Dissemination Policy database which are both used to determine what to disseminate and to whom.

4. System Access Control Policy

Access to information stored on the Dissemination System will be based on the group to which the user belongs. Because not all users should have access to all the information on the system, group-based access control is used. Administrators will be able to access all documents necessary for the performance of their jobs. The internal TCX developers will have read access to specific data on the system. Members of the group that created the documents to be disseminated will have read access to documents they have created. The evaluators will have read access to official documents and data, whereas the collaborators will have read access to engineering releases and jointly-developed documents. The NIST/NSA validators will have sufficient access to evaluation evidence, documents, and code necessary for validating the system evaluation. The customers will only have access to documents that are deemed appropriate per their use licenses. The general public will have the most restricted access to data.

Access control lists will be used to regulate the data available to different groups. Login with a user name and password will be required to access all non-proprietary material. Access to public documents will not require system registration. Configurable audit functions will allow for auditing of user identification, accesses, and other events. The Dissemination System will distribute to users digitally signed versions of unaltered, releasable documents. For specific users, an external signature verification tool will be provided to ensure the integrity of data. However, this tool will not be distributed through the Dissemination System. The system will also distribute the official documents to the validators. These official documents will be signed by the Configuration Manager so that document recipients can verify the version and source of the material. Once validated, this signed version will be releasable to specified user groups via the Dissemination System along with other releasable documents.

5. Document Creation and Viewing

Prior to entering configuration management, documents will be created using XML. When documents cannot be created using XML, for example source code and binaries, then XML document descriptor files will be implemented. This is because non-XML documents cannot have in-line tags with which access control is performed. Versions of documents will be submitted to the Configuration Manager following the standard procedures for the TCX project. Upon release, they will be exported from the configuration management system to the Dissemination System via the Releasing Agent. The documents will include XML tags that specify the user groups allowed to view them. In the case of non-XML data, the document descriptor file will be an XML document with access control tags referring to a specific non-XML file. Document descriptor files will contain tags for all material in one Configuration Item (CI). The CI is the smallest item submitted to Configuration Management for the TCX project. A CI may contain multiple files or be a single file. In both cases, these tags will be bound to all data in order to determine appropriate access levels of users. The tag granularity will be arbitrary but the initial implementation of the Dissemination System will only use document level tags for access control. These access control lists will assure the proper dissemination of data to users. The Dissemination System will adhere to the distribution standards of the XML document tags. Viewing of the XML documents via the Internet will be made possible by XSL transformations. The Dissemination System will implement transformations of XML documents to allow client access.

B. THREAT ANALYSIS

1. Background

A threat analysis must consider the assumptions about the system and its environment, the threats to the system, and the organizational security policies that exist in the target environment. The assumptions are a means to narrow the threats and focus solely on the system being evaluated. All three elements of the threat analysis are created in order to develop a threat model from which requirements can be derived. Threats to the Dissemination System were based on the threats to a standard web server combined with the additional threats resulting from the sensitivity of the information being transmitted. Many of the policies, assumptions, and threats listed below were adapted from the Web Server Protection Profile to fit the TCX dissemination framework [12].

Additionally, the high assurance requirements of the TCX project account for additional threats that would not exist in a low assurance environment. However, before the threats can be considered, assumptions regarding the system must be made. In the context of this work, IT Environment describes the underlying hardware and software upon which the Dissemination System runs. The Dissemination Environment is the Dissemination System and all of the entities described in Figures 1 and 2.

2. Assumptions

Assumptions for the system allow the threats to be narrowed. Table 1 summarizes the conditions that are assumed to exist in the IT Environment and Dissemination Environment.

The physical protection of the system, A.PHYSICAL, secures the system from malicious physical attacks. The physical security measures are assumed to be commensurate with the physical value of the data such that they will help mitigate attacks to the physical system.

A.INTERNAL_PROCESSING	The internal implementation and execution of the Dissemination System's underlying operating system, web server, and cryptographic libraries run as expected.
A.NO_REMOTE_ADMIN	All administrative functions will be performed with access to the physical computer.
A.NO_ROGUE_ADMIN	Administrators are non-hostile, appropriately trained and follow all administrator guidance.
A.PHYSICAL	The Dissemination Environment will provide physical security commensurate with the value of the data being served by the Dissemination System.

Table 1. System Assumptions

Since the administrators effectively have complete control over the system, they are assumed to be non-hostile and trained as described by A.NO_ROGUE_ADMIN. Additionally, when performing administrative tasks, the administrator must be in physical contact with the computer. No remote administration will be performed on the system as described in A.NO_REMOTE_ADMIN.

Finally, the underlying operating system, web server, and cryptographic libraries are assumed to operate correctly. According to A.INTERNAL_PROCESSING, the flow

of data inside the dissemination operating system, web server, and cryptographic libraries will be correct, and the implementation of this software is not flawed. System calls to the operating system will not result in incorrect Dissemination System behavior. Additionally, user interaction with the web server will operate as expected with respect to uniform resource locators (URLs), web content, and vulnerabilities against network protocols (e.g. HTTP, TCP, IP). The cryptographic libraries will properly handle the cryptographic functions and key management.

These assumptions help to mitigate some of the threats to the system. However, there are still many threats that must be taken into account with the development of the Dissemination System.

3. Threats

While the threats are listed below in table format, the most relevant threat to the Dissemination System is T.UNAUTHORIZED_ACCESS. Based on the high assurance nature of the TCX project and the information being disseminated, unauthorized access is the primary concern for the system. The Dissemination System must be able to securely deliver dissemination material to authorized users of the system. Thus the threat of T.ALTERED_DATA must be mitigated by the design. Additionally, T.SYSTEM_COMPROMISE must be mitigated to assure that the data and code of the dissemination system is not inappropriately accessed resulting in the improper dissemination of data to users.

T.ACCIDENTAL_ADMIN_ERROR	The administrator may incorrectly install or configure the Dissemination System which could lead to additional threats and vulnerabilities.
T.ACCIDENTAL_AUDIT_ COMPROMISE	A user may accidentally view, modify or delete audit records resulting in a user's actions to be masked.
T.ALTERED_DATA	The end user version of dissemination data may differ from the master version of the releasable data maintained by the Dissemination System thus resulting in corrupted delivery.
T.MASQUERADE	A foreign entity may masquerade as the Dissemination System resulting in misrepresentation of the Dissemination System's controlled dissemination data.

T.POOR_DESIGN	Unintentional errors in the requirements specification or design of the Dissemination Application may occur, leading to flaws that may be exploited by a casually mischievous user or program.
T.POOR_IMPLEMENTATION	Unintentional errors in implementation of the Dissemination Application design may occur, leading to flaws that may be exploited by a casually mischievous user or program.
T.POOR_TEST	Lack of or insufficient tests of the Dissemination Application to demonstrate that all security functions operate correctly may result in incorrect behavior being undiscovered thereby causing potential security vulnerabilities.
T.ROGUE_ENTITY	The Dissemination Environment entities, other than the Dissemination System, may not be trustworthy thus resulting in the improper dissemination of data.
T.SYSTEM_COMPROMISE	The Dissemination System data and/or code may be inappropriately accessed resulting in the improper dissemination of data to users.
T.UNATTENDED_SESSION	A user may gain unauthorized access to an unattended administrator session.
T.UNAUTHORIZED_ACCESS	A user may gain access to dissemination data for which the user is not authorized according to the access control attributes of the data.
T.UNIDENTIFIED_ACTIONS	The administrator may not have the ability to notice potential security violations, thus limiting the administrator's ability to identify and take action against a possible security breach.

Table 2. System Threats

The administrator threat of T.ACCIDENTAL_ADMIN_ERROR is the fourth greatest risk to the Dissemination System. The incorrect installation of the system may include incorrect access control lists on files or incorrect installation or configuring of the Dissemination System processes. This threat can lead to other threats already stated. The administrator must also prevent users gaining access to unattended administrator sessions. This threat, T.UNATTENDED_SESSION could result in a malicious user corrupting the entire system. The administrator must also be aware of the threat of

T.UNIDENTIFIED_ACTIONS and realize that some user activity could possibly be hidden.

While auditing will be performed on the system to enhance the security, this adds the threat of T.ACCIDENTAL_AUDIT_COMPROMISE. Viewing, modifying, or deleting of the audit records by clients or processes could cause a user's actions to be masked or not recorded. This would create an unwanted circumstance that could be potentially threatening to the system.

The Dissemination Environment consists of multiple entities that provide technical measures and information necessary for secure dissemination of material. If these entities are not trustworthy then the improper dissemination of data to users could result (T.ROGUE_ENTITY).

The threat of a foreign entity masquerading as the Dissemination System may result in the misrepresentation of the Dissemination System's controlled dissemination data as described by T.MASQUERADE.

T.POOR_DESIGN, T.POOR_IMPLEMENTATION, and T.POOR_TEST are all relevant threats to a newly developed system. If the design or implementation is incomplete or poorly done, then potential threats become a reality on the system. Additionally, poor testing of the system once implemented can leave security vulnerabilities on the system.

The requirements specification for the Dissemination System will address the threats to mitigate the risks posed by the threats discussed above. But first, policies regarding system use must be articulated. This will help to identify which threats are valid and which should never occur on the system.

C. ORGANIZATIONAL SECURITY POLICIES

The policies applied to the system set the limits for system use. They define the acceptable use of the system. Mainly, they are security policies for the system and thus help to maintain the assurance of the data.

P.ACCESS_BANNER	The Dissemination System will present a banner to all users describing restrictions of use, legal agreements, or any other

	appropriate information to which users consent by accessing the system.
P.ACCOUNTABILITY	The authorized users of the system shall be held accountable for their actions on the system.
P.DATA_MARKING	All dissemination data will be properly marked with access control attributes.

Table 3. Security Policies

Users must realize who controls the system and the acceptable activities allowed on the system. Additionally, they must understand that they are held responsible for their actions while using the system. Therefore the policies of P.ACCESS_BANNER and P.ACCOUNTABILITY are enacted on the system.

In order to process the data correctly by the Dissemination System, P.DATA_MARKING must be used. Data must be marked or tagged properly in order to correctly disseminate it. These data markings include the access control attributes which are pivotal to the proper dissemination of data.

D. SECURITY OBJECTIVES

To address the threats, organizational security policies and assumptions described above, a set of security objectives for both the Dissemination System and the environment are required. The Dissemination System security objectives are developed to mitigate the threats and implement the policies of the Dissemination System. The security objectives for the environment are created to address the assumptions about the environment in which the Dissemination System operates.

1. Dissemination System Security Objectives

The security objectives for the Dissemination System are summarized in Table 4.

The objective O.ACCESS requires the Dissemination System to only disseminate information in accordance with the access control policy and ensure that dissemination material is properly distributed only to authorized users. The authorized users are provided with confidence that the received material is from the TCX Dissemination System and matches the master version maintained on the Dissemination System (O.USER_CONFIDENCE and O.SECURE_DELIVERY). Prior to allowing the users to

access data, the Dissemination System will inform the users of proper user behavior, website policies, and required authentication methods (O.DISPLAY_BANNER and O.USER_GUIDANCE). The Dissemination System will also provide mechanisms that control a user's logical access to the system (O.SYSTEM_ACCESS).

Multiple objectives exist to address the generation of audit data, protection of the audit trail, and analysis of audit records. These objectives are O.AUDIT_GENERATION, O.AUDIT_PROTECTION, and O.AUDIT_REVIEW. Additionally, time stamps will be utilized by the audit mechanisms for accountability purposes as described by the O.TIME_STAMPS objective.

To ensure that the administrator has all information necessary for securely administering the Dissemination System, administrative guidance will be provided (O.ADMIN_GUIDANCE).

O.ACCESS	The Dissemination System will ensure that users gain only authorized access to resources that it controls.
O.ADMIN_GUIDANCE	The Dissemination System will provide administrators with the necessary information for secure management.
O.AUDIT_GENERATION	The Dissemination System will provide the capability to detect and create records of security-relevant events associated with users.
O.AUDIT_PROTECTION	The Dissemination System will provide the capability to protect audit information.
O.AUDIT_REVIEW	The Dissemination System will provide the capability to view audit information.
O.CONGIFURATION_MANAGEMENT	The configuration of and all changes to the Dissemination System will be tracked and

	controlled throughout the Dissemination System's development.
O.DISPLAY_BANNER	The Dissemination System will display an advisory warning regarding use of the Dissemination System.
O.DOCUMENTED_DESIGN	The design of the Dissemination Application will be adequately and accurately documented.
O.PARTIAL_FUNCTIONAL_TESTING	The Dissemination System will undergo some security functional testing that demonstrates that it satisfies some of its security functional requirements.
O.SECURE_DELIVERY	The Dissemination System will provide mechanisms for users to verify that any data disseminated matches the master version maintained by the Dissemination System.
O.SOUND_IMPLEMENTION	The implementation of the Dissemination Application will be an accurate instantiation of its design.
O.SYSTEM_ACCESS	The Dissemination System will provide mechanisms that control a user's logical access to it.
O.TIME_STAMPS	The Dissemination System will use time stamps for accountability purposes.
O.USER_CONFIDENCE	The Dissemination System will provide mechanisms that permit end users to have confidence that received controlled-access data comes from the Dissemination System.

O.USER_GUIDANCE	The Dissemination System will provide users with the necessary information for secure data access.
O.VULNERABILITY_ANALYSIS	The Dissemination Application will undergo some vulnerability analysis to demonstrate the design and implementation do not contain any obvious flaws.

Table 4. Functional Security Objectives

The Dissemination System design will be sufficiently and correctly documented to ensure all functional requirements are satisfied (O.DOCUMENTED_DESIGN). The objective O.CONFIGURATION_MANAGEMENT requires all developmental material (both original and subsequent changes) be maintained by Configuration Management. The implementation of the Dissemination Application will be an accurate instantiation of the design specification as specified by O.SOUND_IMPLEMENTATION. Following the implementation, a partial functional test and vulnerability analysis will be conducted to assure that the system satisfies a subset of the functional requirements and does not contain any obvious flaws (O.PARTIAL_FUNCTIONAL_TESTING and O.VULNERABILITY_ANALYSIS).

2. Security Objectives for the Environment

The security objectives state the goals that the IT Environment and the Dissemination Environment must address. The objectives for the environment are summarized in Table 5.

OE.DATA_MARKING	All dissemination data will be properly marked with access control attributes by the Dissemination Environment.
OE.INTERNAL_PROCESSING	The internal implementation and execution of the Dissemination System's underlying operating system, web server, and cryptographic libraries will run as expected.

OE.MANAGE	The IT Environment will provide all the functions and facilities necessary to support the administrators in their management of the security of the Dissemination System, and restrict these functions and facilities from unauthorized use.
OE.NO_REMOTE_ADMIN	The IT Environment will provide only local capabilities for Dissemination System administration.
OE.NO_ROGUE_ADMIN	The Dissemination Environment shall ensure that administrators are non-hostile, appropriately trained, and follow all administrator guidance.
OE.NO_ROGUE_ENTITY	The Dissemination Environment shall ensure that all of its entities are trustworthy and protect both the dissemination data and the data they generate to support secure distribution.
OE.PHYSICAL	Physical security, commensurate with the value of the Dissemination System and the data it contains, will be provided by the Dissemination Environment.
OE.REGISTERED_USERS	The Dissemination Environment will provide a mechanism for users to register prior to accessing non-public material on the Dissemination System.
OE.RELIABLE_TIME_STAMP	The IT Environment will provide reliable time stamps.
OE.SYSTEM_PROTECTION	The IT Environment will provide sufficient mechanisms to protect the Dissemination System's data and memory during storage and execution.

Table 5. Operational Environment Security Objectives

The IT Environment will provide physical security to protect the Dissemination System and its data (OE.PHYSICAL). In addition to the physical security, the IT Environment must also provide sufficient mechanisms to protect the data and memory on

the Dissemination System during its storage and execution (OE.SYSTEM_PROTECTION). The Dissemination System administrators shall be non-hostile, appropriately trained, and follow all administrator guidance as specified by the OE.NO_ROGUE_ADMIN objective. The IT Environment will ensure that remote administration is not permitted on the Dissemination System thus fulfilling OE.NO_REMOTE_ADMIN.

The IT Environment upon which the Dissemination System is built must assure that the operating system, web server, and cryptographic libraries function as expected (OE.INTERNAL_PROCESSING). OE.RELIABLE_TIME_STAMP states that the IT Environment must also provide reliable time stamps for use by the Dissemination System. The administrators of the Dissemination System will be provided with all the functions and facilities to securely administer the Dissemination System, as described in OE.MANAGE. It is also the responsibility of the Dissemination Environment to assure that the entities within its domain are trustworthy as defined by OE.NO_ROGUE_ENTITY. The Dissemination Environment must provide a mechanism for users to register for non-public access to data controlled by the Dissemination System. This is objective is defined as OE.REGISTERED_USERS.

To properly disseminate data, the data must first be marked with dissemination access control attributes. These access control attributes will be applied to dissemination data by the Dissemination Environment (OE.DATA_MARKING).

E. SYSTEM REQUIREMENTS

1. Background

The requirements for the Dissemination System can be split into two categories: requirements for trusted delivery and requirements for the dissemination of data. Each threat must be mapped to one or more functional and/or assurance requirements whose purpose is to mitigate that threat. However, all requirements will not map to a single threat.

2. Secure Delivery

The Dissemination System must be able to deliver data to evaluators in addition to the dissemination of project material across the Internet. There are two parts to fulfilling the secure delivery requirement of the system. First, the on-site versions of

documents must match the master version. To assure this, documents will be signed so that their authenticity can be verified. Additionally, an external signature verification tool shall be implemented to permit verification of the signatures on documents. This tool must also be responsible for verifying the seal of the TCX kernel being disseminated. Second, if documents are modified after delivery, then they are not considered an evaluated version. The secure delivery requirement can guarantee that upon delivery all documents shall be signed, but cannot protect the documents further. End users may use the external signature verification tool to check signatures after downloading data from the system. This shall be the users' method of verifying that unmodified, complete versions of documents were received. These two requirements mitigate the T.ALTERED_DATA threat to the dissemination web server by guaranteeing that the signed version is the only one to be disseminated to users. This includes but is not limited to: XML documents, source code, and binaries.

3. Dissemination of Data/Documents

The dissemination of data imposes additional requirements necessary to mitigate the threats to the system. Requirements for trusted delivery are also required for the dissemination of data, but will not be listed again.

a. Identification and Authentication with Audit

All users are required to register before accessing proprietary information on the Dissemination System. This will be performed external to the Dissemination System. Each registered user of the system is required to have a unique user identification (user ID) and a password. The user is also assigned a user group which will be used to determine more granular access. The user is required to login to the system for access to non-public documents. The login process requires the user to enter his or her user ID and password to authenticate to the server.

Audit logging will be required in order to monitor the identification and authentication process. Security critical events shall be audited to include: dissemination of new documents to the server, changes to existing documents, and changing of user access levels. Additionally, the login and logoff process will be audited. The security policy for the system will describe all necessary audit logging for the system and will include the aforementioned.

b. Group-based Access Control

The group-based access control allows all users to be placed in a group, or a collection of groups, for access to data. These groups include: administrator, collaborator, customer, developer, evaluator, NIST/NSA validators, and public. All documents shall contain an access control list specifying which groups have access to that particular document. These access control lists shall be in the form of XML tags embedded in the document. As stated previously, non-XML documents shall have access control tags in XML document descriptor files.

c. XML Binding and XSL Transformations

All documents disseminated shall be XML documents or have an XML document descriptor file. XML access control tags will be either embedded in the XML document or the document descriptor file. These tags will be processed by the Dissemination System in order to disseminate material to the proper users. Note that if the tags of the document are modified, the document will no longer have a valid signature.

To display documents via a standard web browser, the XML documents shall require XSL transformations. This will allow the conversion of XML to HTML and simple viewing across the Internet. The XML tags shall be crucial in formatting the documents, but will not be visible when viewed by a web browser. Additionally, for integrity verification the HTML document will have a web link to download the source XML document containing its digital signature.

F. CONCLUSION

The TCX project is most useful by being disseminated across the Internet so that others can learn from it. The Dissemination System has specific assumptions and threats that must be dealt with in order to maintain the security of information. Additionally, organizational security policies help to clarify the actual use of the system. The assumptions, threats, and policies help to determine the requirements for the Dissemination System. With specific requirements, the design and future implementation will be better developed to obtain maximum functionality and security on the Dissemination System. This all contributes to the fulfillment of the high level system functional description while mitigating the threats through specific requirements.

IV. SECURITY REQUIREMENTS

This chapter contains the security requirements for the Dissemination System and the Dissemination Application as driven by the assumptions, threats, policies, and objectives. Although these requirements were developed using the Common Criteria [2] as a framework, they do not follow the Common Criteria constructs and language. Additionally, the requirements are broken into two categories: security functional requirements and security assurance requirements. Next, the mapping of the security objectives to the threats, policies, and assumptions, and the mapping of the requirements to the security objectives are provided.

A. DISSEMINATION SYSTEM SECURITY FUNCTIONAL REQUIREMENTS

1. Dissemination System Audit

1.1 The Dissemination System shall have configurable auditing capabilities. Audit levels shall be hierarchical from the least amount of information to the most. The Dissemination System shall support the following audit levels: alert, critical, error, warning, notice, information, and debugging. All audited events shall be recorded.

1.2 The date and time of the event, number of bytes sent to the server, the remote host name or IP address, type of event, user identity (if applicable), and the outcome (success or failure) shall be recorded.

1.3 The audit records generated by the Dissemination System shall be in a format that can be parsed.

1.4 An authorized administrator shall be able to select the amount of time between audit log rotations. This shall be configurable for daily, weekly, or monthly rotations. Additionally, the administrator shall be able to specify how many rotated logs are kept on the system before being archived.

2. Dissemination System Communication

2.1 The Dissemination System shall employ cryptographic functionality to provide a secure connection between the Dissemination System and the users.

3. Dissemination System Cryptography

3.1 The Dissemination System shall use a digital certificate signed by an authorized Certificate Authority for authenticating itself to the users.

4. Dissemination System User Data Protection

4.1 The Dissemination System shall enforce the access control policy on all registered users and data on the system. This policy shall be enforced based on the user ID and group membership for the data requested.

5. Dissemination System Identification and Authentication

5.1 The Dissemination System shall ensure that users are identified and authenticated in order to associate them with the proper security attributes while accessing data. Security attributes shall include but are not limited to the user's identity and the group(s) to which that user belongs.

5.2 The Dissemination System shall authenticate registered users based on their user ID and password.

5.3 The Dissemination System shall authenticate users prior to allowing access to any non-public documents on the Dissemination System.

6. Dissemination System Access

6.1 The Dissemination System shall clearly display an access banner describing the restrictions of use, legal agreements, or any other appropriate information to which users consent by accessing the system.

B. DISSEMINATION SYSTEM SECURITY ASSURANCE REQUIREMENTS

1. Dissemination System Configuration Management

1.1 The Dissemination System's third party software and documentation including configuration files shall be maintained under Configuration Management.

2. Dissemination System Guidance Documents

2.1 The user guidance shall describe the interaction between the user and the Dissemination System for proper retrieval of releasable data.

2.2 The user guidance shall clearly present all user responsibilities necessary for secure use of the Dissemination System, including those related to assumptions regarding user behavior found in the access banner.

2.3 The administrative guidance shall describe the procedures and technical measures necessary to restrict physical access to the system.

2.4 The administrative guidance shall cover configuration, maintenance, and administration of the Dissemination System in a secure manner. The guidance is intended to help administrators understand the security functions of the Dissemination System, including both those functions that require the administrator to perform security-critical actions and those functions that provide security-critical information to the administrator [2].

2.5 The administrative guidance shall describe the functions and interfaces available to the administrator in addition to how to manage the Dissemination System in a secure manner [2].

2.6 The administrative guidance shall describe all security requirements for the Dissemination Environment that are relevant to the administrator and the Dissemination System.

3. Dissemination System Testing

3.1 The Dissemination System test plan shall cover expected usage of the Dissemination System in addition to limited testing of unexpected situations. This partial functional testing shall ensure that the Dissemination System properly performs functions required for dissemination.

C. DISSEMINATION APPLICATION SECURITY FUNCTIONAL REQUIREMENTS

1. Dissemination Application Audit

1.1 The Dissemination Application shall have configurable auditing capabilities. All audited events shall be recorded.

1.2 The date and time of the event, type of event, user identity (if applicable), and the outcome (success or failure) shall be recorded.

1.3 The audit records generated by the Dissemination Application shall be in a format that can be parsed.

2. Dissemination Application User Data Protection

2.1 The Dissemination Application shall enforce the policy based on the following dissemination file attributes: file type marking, file sensitivity marking.

2.2 The Dissemination Application shall enforce all dissemination access control mechanisms on all dissemination material present on the server.

2.3 The Dissemination Application shall not modify the digital signatures placed on the dissemination material by the Configuration Management system.

2.4 The Dissemination Application shall redact each releasable document in accordance with the dissemination policy.

D. DISSEMINATION APPLICATION SECURITY ASSURANCE REQUIREMENTS

1. Dissemination Application Configuration Management

1.1 The Dissemination Application documents (e.g. functional specification) and software shall be archived using the CM process.

2. Dissemination Application Operation

2.1 Installation and startup procedures shall be documented within the administrative guidance to ensure that the Dissemination Application has been installed and started in a secure manner, as intended by the developer.

3. Dissemination Application Development

3.1 An informal high level design specification shall be developed for the Dissemination Application.

3.2 The design of the Dissemination Application shall meet the functional requirements.

3.3 The Dissemination Application shall be implemented in accordance with the design.

4. Dissemination Application Guidance Documents

4.1 The administrative guidance shall cover configuration, maintenance, and administration of the Dissemination Application in a secure manner.

4.2 The administrative guidance shall describe the functions and interfaces available to the administrator in addition to how to manage the Dissemination Application in a secure manner [2].

4.3 The administrative guidance shall describe all security requirements for the Dissemination System that are relevant to the administrator and the Dissemination Application.

5. Dissemination Application Life Cycle Support

5.1 The Dissemination Application shall follow the same spiral life cycle model and procedures as the TCX project.

6. Dissemination Application Testing and Vulnerability Assessment

6.1 The developer shall develop a test plan to cover expected usage of the Dissemination Application in addition to limited testing of unexpected situations. This partial functional testing shall ensure that the Dissemination Application properly handles access to releasable items while maintaining its own configuration [2].

6.2 To help mitigate misuse of the Dissemination Application the guidance documentation shall be complete, clear, consistent, and reasonable. It shall list the assumptions about the environment, and requirements for external security measures [2].

6.3 The developer shall perform a vulnerability assessment of the Dissemination Application along with provision of vulnerability analysis documentation [2].

E. IT ENVIRONMENT SECURITY FUNCTIONAL REQUIREMENTS

1. IT Environment Security Management

1.1 The IT Environment shall restrict the ability to perform the following functions to the authorized administrator: enable, disable, and modify the audit settings; backup and restore audit records; adjust the configuration parameters; and modification of any databases used by the Dissemination Application.

1.2 The IT Environment shall restrict the ability to modify the security attributes of both data and users of the system to the authorized administrator.

2. IT Environment Access

2.1 The IT Environment shall lock a local interactive administrator session after a specified period of inactivity by disabling system access to data and display devices other than the ability to unlock the session. The IT Environment shall require re-authentication by the administrative user prior to unlocking the session.

3. IT Environment Data Protection

3.1 The IT Environment shall restrict the ability to create or modify webpage content to authorized administrators.

3.2 The IT Environment shall be capable of limiting the ability to create or modify server executable content [2].

3.3 The IT Environment shall protect the Dissemination System's private key from unauthorized modification and viewing.

4. IT Environment Audit

4.1 The IT Environment shall restrict all non-administrative users of the Dissemination System from reading from and writing to the audit trail.

4.2 The IT Environment shall be able to provide time stamps for its own use.

5. IT Environment Identification and Authentication

5.1 The IT Environment shall ensure that users are identified and authenticated in order to associate them with the proper security attributes while accessing data. Security attributes shall include but are not limited to the user's identity and the group(s) to which that user belongs.

F. DISSEMINATION ENVIRONMENT SECURITY FUNCTIONAL REQUIREMENTS

1. Dissemination Environment Security Management

1.1 The Dissemination Environment shall provide a mechanism for users to register prior to accessing non-public material on the Dissemination System.

2. Dissemination Environment User Data Protection

2.1 The Dissemination Environment shall properly mark all dissemination data with access control attributes.

3. Dissemination Environment Protection

3.1 The Dissemination Environment shall ensure that all data generated by its entities required for proper dissemination is protected in situ and during transit.

G. REQUIREMENTS MAPPING

All assumptions, threats, policies, and security objectives were previously defined in Chapter III. Using the Common Criteria methodology, this section maps the threats and policies to the security objectives of the Dissemination System. Then the

assumptions are mapped to the security objectives of the environment. Finally, the mapping of the requirements to the security objectives is presented. The rationale in this section was adapted from the Web Server Protection Profile and the Separation Kernel Protection Profile in addition to the Basic Robustness Consistency Instruction Manual [2, 9, 12].

1. Threat and Policy Mapping

Table 6 contains the mapping of the objectives to the threats and policies and provides the rationale for how the objectives mitigate the threats and implement the policies.

Threat/Policy	Objectives Addressing the Threats and Policies	Rationale
T.ACCIDENTAL_ADMIN_ERROR: The administrator may incorrectly install or configure the Dissemination System which could lead to additional threats and vulnerabilities.	O.ADMIN_GUIDANCE: The Dissemination System will provide administrators with the necessary information for secure management.	O.ADMIN_GUIDANCE helps to mitigate this threat by requiring the system administrators to have guidance that instructs them how to administer the system in a secure manner. Having this guidance helps to reduce the mistakes that an administrator might make that could cause an insecure configuration.
T.ACCIDENTAL_AUDIT_COMPROMISE: A user may accidentally view, modify or delete audit records resulting in a user's actions to be masked.	O.AUDIT_PROTECTION: The Dissemination System will provide the capability to protect audit information.	O.AUDIT_PROTECTION requires that this threat be mitigated by controlling access to the audit trail. Only the system administrator is provided read/write access to the audit trail.
T.ALTERED_DATA: The end user version of dissemination data may differ from the master version of the releasable data maintained by the Dissemination System thus resulting in	O.SECURE_DELIVERY: The Dissemination System will provide mechanisms for users to verify that any data disseminated matches the master version maintained by the Dissemination	O.SECURE_DELIVERY requires that this threat be mitigated by providing a mechanism to allow the end user to verify the integrity of the dissemination material through

corrupted delivery.	System. O.SYSTEM_ACCESS: The Dissemination System will provide mechanisms that control a user's logical access to it.	the use of digital signatures. O.SYSTEM_ACCESS requires that mechanisms are implemented that protect the master version of the releasable data from unauthorized access.
T.MASQUERADE: A foreign entity may masquerade as the Dissemination System resulting in misrepresentation of the Dissemination System's controlled dissemination data.	O.USER_CONFIDENCE: The Dissemination System will provide mechanisms that permit end users to have confidence that received controlled-access data comes from the Dissemination System. O.USER_GUIDANCE: The Dissemination System will provide users with the necessary information for secure data access.	O.USER_CONFIDENCE requires that this threat be mitigated by server authentication mechanisms that allow the Dissemination System to authenticate itself to the client prior to users accessing controlled dissemination data. O.USER_GUIDANCE helps to mitigate this threat by requiring the user guidance to describe the server authentication method and how to configure the client to authenticate the server.
T.POOR_DESIGN: Unintentional errors in the requirements specification or design of the Dissemination Application may occur, leading to flaws that may be exploited by a casually mischievous user or program.	O.DOCUMENTED_DESIGN: The design of the Dissemination Application will be adequately and accurately documented. O.VULNERABILITY_ANALYSIS: The Dissemination Application will undergo some vulnerability analysis to demonstrate the design and implementation do not contain any obvious flaws. O.CONFIGURATION_MANAGEMENT: The configuration of and all changes to the Dissemination System will be	O.DOCUMENTED_DESIGN requires that the design of the Dissemination Application be documented, permitting review for poor design and, thus, mitigating this threat. O.VULNERABILITY_ANALYSIS requires that the design of the Dissemination Application be analyzed for design flaws. To mitigate this threat, O.CONFIGURATION_MANAGEMENT requires that changes to the system design be tracked, thus mitigating the threat that

	tracked and controlled throughout the Dissemination System's development.	changes could result in poor design of the system.
T.POOR_IMPLEMENTATION: Unintentional errors in implementation of the Dissemination Application design may occur, leading to flaws that may be exploited by a casually mischievous user or program.	O.SOUND_IMPLEMENTATION: The implementation of the Dissemination Application will be an accurate instantiation of its design. O.PARTIAL_FUNCTIONAL_TESTING: The Dissemination System will undergo some security functional testing that demonstrates that it satisfies some of its security functional requirements. O.VULNERABILITY_ANALYSIS: The Dissemination Application will undergo some vulnerability analysis to demonstrate the design and implementation do not contain any obvious flaws. O.CONFIGURATION_MANAGEMENT: The configuration of and all changes to the Dissemination System will be tracked and controlled throughout the Dissemination System's development.	To mitigate this threat, O.SOUND_IMPLEMENTATION requires that the implementation be an accurate representation of the design. To mitigate this threat, O.PARTIAL_FUNCTIONAL_TESTING requires testing that increases the likelihood that any errors that do exist in the implementation will be discovered. O.VULNERABILITY_ANALYSIS mitigates this threat by requiring the reduction of errors in the implementation that may not be discovered during functional testing. Ambiguous design documentation and the fact that exhaustive testing of the external interfaces is not required may leave flaws in the implementation undiscovered in functional testing. O.CONFIGURATION_MANAGEMENT helps to mitigate this threat by requiring that all modifications to the Dissemination Application be tracked thus reducing the number of potential exploits.
T.POOR_TEST: Lack of or	O.PARTIAL_FUNCTIONAL_T	O.PARTIAL_FUNCTIONAL_T

insufficient tests of the Dissemination Application to demonstrate that all security functions operate correctly may result in incorrect behavior being undiscovered thereby causing potential security vulnerabilities.	ESTING: The Dissemination System will undergo some security functional testing that demonstrates that it satisfies some of its security functional requirements. O.VULNERABILITY_ANALYSIS: The Dissemination Application will undergo some vulnerability analysis to demonstrate the design and implementation do not contain any obvious flaws.	ESTING helps to mitigate this threat by requiring testing that increases the likelihood that any errors that do exist in the implementation will be discovered. O.VULNERABILITY_ANALYSIS addresses this concern by requiring a vulnerability analysis be performed in conjunction with testing that goes beyond functional testing. This objective provides a measure of confidence that the Dissemination Application does not contain security flaws that may not be identified through functional testing.
T.ROGUE_ENTITY: The Dissemination Environment entities, other than the Dissemination System, may not be trustworthy thus resulting in the improper dissemination of data.	OE.NO_ROGUE_ENTITY: The Dissemination Environment shall ensure that all of its entities are trustworthy and protect both the dissemination data and the data they generate to support secure distribution.	OE.NO_ROGUE_ENTITY addresses this threat by requiring that the Dissemination Environment ensure that all of its entities are trustworthy and protect the data they generate.
T.SYSTEM_COMPROMISE: The Dissemination System data and/or code may be inappropriately accessed resulting in the improper dissemination of data to users.	OE.SYSTEM_PROTECTION: The IT Environment will provide sufficient mechanisms to protect the Dissemination System's data and memory during storage and execution. OE.MANAGE: The IT Environment will provide all the functions and facilities necessary to support the administrators in their management of the security	OE.SYSTEM_PROTECTION partially mitigates this threat by requiring access controls on the data and code of the system thus protecting data and code during storage and execution. OE.MANAGE partially mitigates this threat by requiring that the administrator be provided with the functions necessary to control access to data and code on the

	of the Dissemination System, and restrict these functions and facilities from unauthorized use. OE.INTERNAL_PROCESSING: The internal implementation and execution of the Dissemination System's underlying operation system, web server, and cryptographic libraries will run as expected.	Dissemination System. OE.INTERNAL_PROCESSING requires partial mitigation of this threat by assuming that the underlying operating system, web server, and cryptographic libraries operate as expected.
T.UNATTENDED_SESSION: A user may gain unauthorized access to an unattended administrator session.	OE.MANAGE: The IT Environment will provide all the functions and facilities necessary to support the administrators in their management of the security of the Dissemination System, and restrict these functions and facilities from unauthorized use.	OE.MANAGE helps to mitigate this threat by requiring mechanisms that place controls on user's sessions to be implemented. Local administrator's sessions are locked after a system administrator defined time period of inactivity. Locking the local administrator's session reduces the opportunity of someone gaining unauthorized access to the session when the console is unattended.
T.UNAUTHORIZED_ACCESS: A user may gain access to dissemination data for which the user is not authorized according to the access control attributes of the data.	O.ACCESS: The Dissemination System will ensure that users gain only authorized access to resources that it controls. O.SYSTEM_ACCESS: The Dissemination System will provide mechanisms that control a user's logical access to it. OE.DATA_MARKING: All dissemination data will be properly marked with access	To partially mitigate this threat, O.ACCESS requires that all access to dissemination data be strictly controlled according to the access control attributes for each data item. To partially mitigate this threat, O.SYSTEM_ACCESS requires mechanisms that identify and authenticate the user prior to the user's access to dissemination data.

	control attributes. OE.REGISTERED_USERS: The Dissemination Environment will provide a mechanism for users to register prior to accessing non-public material on the Dissemination System.	OE.DATA_MARKING requires that dissemination data be properly marked with access control attributes. The mechanism required by OE.REGISTERED_USERS will provide the user authentication data that will be used to authenticate users. This will help mitigate the threat of unauthorized access.
T.UNIDENTIFIED_ACTIONS: The administrator may not have the ability to notice potential security violations, thus limiting the administrator's ability to identify and take action against a possible security breach.	O.AUDIT_REVIEW: The Dissemination System will provide the capability to view audit information. O.AUDIT_GENERATION: The Dissemination System will provide the capability to detect and create records of security-relevant events associated with users. O.TIME_STAMPS: The Dissemination System will use time stamps for accountability purposes. OE.RELIABLE_TIME_STAMP: The IT Environment will provide reliable time stamps.	O.AUDIT_REVIEW helps to mitigate this threat by requiring that the system administrator be provided with the capability to review audit data for activity that could indicate a potential security violation. O.AUDIT_GENERATION helps to mitigate this threat requiring that auditable events be recorded for later review. O.TIME_STAMPS helps to mitigate this threat by requiring that audit records have correct time stamps. OE.RELIABLE_TIME_STAMP helps to mitigate this threat by requiring that the time stamp functions used by the Dissemination System be provided.
P.ACCESS_BANNER: The Dissemination System will present a banner to all users	O.DISPLAY_BANNER: The Dissemination System will display an advisory warning	O.DISPLAY_BANNER satisfies this policy by requiring that the Dissemination System display an

describing restrictions of use, legal agreements, or any other appropriate information to which users consent by accessing the system.	regarding use of the Dissemination System.	administrator configurable banner that provides all interactive users with a warning about the unauthorized use of the Dissemination System and its government ownership.
P.ACCOUNTABILITY: The authorized users of the system shall be held accountable for their actions on the system.	O.AUDIT_GENERATION: The Dissemination System will provide the capability to detect and create records of security-relevant events associated with users. O.TIME_STAMPS: The Dissemination System will use time stamps for accountability purposes. O.USER_GUIDANCE: The Dissemination System will provide users with the necessary information for secure data access. O.DISPLAY_BANNER: The Dissemination System will display an advisory warning regarding use of the Dissemination System. OE.RELIABLE_TIME_STAMP: The IT Environment will provide reliable time stamps. O.SYSTEM_ACCESS: The Dissemination System will provide mechanisms that control a user's logical access to it.	O.AUDIT_GENERATION addresses this policy by requiring that the system administrator be provided with a means of assuring that users are accountable for their actions. O.TIME_STAMPS addresses this policy by requiring that time stamps be provided to trace user actions for which they are accountable. O.USER_GUIDANCE addresses this policy by requiring that all the information necessary for users to securely access dissemination material be provided. User guidance inherently includes acceptable activities that are allowed on the system. O.DISPLAY_BANNER addresses this policy by requiring warnings that users are accountable for their actions on the system and that are displayed at every access to the system. OE.RELIABLE_TIME_STAMP addresses this policy by requiring that time stamp functions be

		provided for use by the Dissemination System. O.SYSTEM_ACCESS addresses this policy by requiring that identification and authentication mechanisms be provided to help implement user accountability.
P.DATA_MARKING: All dissemination data will be properly marked with access control attributes.	OE.DATA_MARKING: All dissemination data will be properly marked with access control attributes by the Dissemination Environment.	OE.DATA_MARKING addresses this policy by requiring that the Dissemination Environment provide data markings for access control.

Table 6. Threat and Policy Mapping

2. Assumption Mapping

Table 7 contains the mapping between the assumptions and the environment security objectives and the rationale for how the objectives address the assumptions.

Assumptions	Objectives Addressing the Assumptions	Rationale
A.INTERNAL_PROCESSING: The internal implementation and execution of the Dissemination System's underlying operating system, web server, and cryptographic libraries run as expected.	OE.INTERNAL_PROCESSING: The internal implementation and execution of the Dissemination System's underlying operating system, web server, and cryptographic libraries will run as expected.	OE.INTERNAL_PROCESSING addresses this assumption by requiring that the internal implementation and execution of the Dissemination System's underlying operating system, web server, and cryptographic libraries run as expected.
A.NO_ROGUE_ADMIN: Administrators are non-hostile, appropriately trained and follow all administrator guidance.	OE.NO_ROGUE_ADMIN: The Dissemination Environment shall ensure that administrators are non-hostile, appropriately trained, and follow all administrator guidance.	OE.NO_ROGUE_ADMIN addresses this assumption by requiring that the Dissemination Environment ensure that administrators are non-hostile, appropriately trained, and follow all administrator guidance.
A.NO_REMOTE_ADMIN: All	OE.NO_REMOTE_ADMIN:	OE.NO_REMOTE_ADMIN

administrative functions will be performed with access to the physical computer.	The IT Environment will provide only local capabilities for Dissemination System administration..	addresses this assumption by requiring that the IT Environment provide only local capabilities for Dissemination System administration.
A.PHYSICAL: The Dissemination Environment will provide physical security commensurate with the value of the data being served by the Dissemination System.	OE.PHYSICAL: Physical security, commensurate with the value of the Dissemination System and the data it contains, will be provided by the Dissemination Environment.	OE.PHYSICAL addresses this assumption by requiring that physical security, commensurate with the value of the Dissemination System and the data it contains, be provided by the Dissemination Environment.

Table 7. Assumption Mapping

3. Requirements Mapping

This section maps the requirements to the objectives that they support and explains how the requirements implement the objectives. The requirements were defined in sections A through F of this chapter. Table 8 consists of two parts. Part I maps the requirements to the security objectives of the Dissemination System. Part II shows the mapping between the environment objectives and the requirements implemented by the Dissemination Environment and the IT Environment.

PART I: SYSTEM REQUIREMENTS MAPPING
O.ACCESS: The Dissemination System will ensure that users gain only authorized access to resources that it controls.
A.4.1 The Dissemination System shall enforce the access control policy on all registered users and data on the system. This policy shall be enforced based on the user ID and group membership for the data requested. C.2.1 The Dissemination Application shall enforce the policy based on the following dissemination file attributes: file type marking, file sensitivity marking. C.2.2 The Dissemination Application shall enforce all dissemination access control mechanisms on all dissemination material present on the server. C.2.4 The Dissemination Application shall redact each releasable document in accordance with the

dissemination policy.
O.ADMIN_GUIDANCE: The Dissemination System will provide administrators with the necessary information for secure management.
B.2.3 The administrative guidance shall describe the procedures and technical measures necessary to restrict physical access to the system. B.2.4 The administrative guidance shall cover configuration, maintenance, and administration of the Dissemination System in a secure manner. The guidance is intended to help administrators understand the security functions of the Dissemination System, including both those functions that require the administrator to perform security-critical actions and those functions that provide security-critical information to the administrator. B.2.5 The administrative guidance shall describe the functions and interfaces available to the administrator in addition to how to manage the Dissemination System in a secure manner. B.2.6 The administrative guidance shall describe all security requirements for the Dissemination Environment that are relevant to the administrator and the Dissemination System. D.2.1 Installation and startup procedures shall be documented within the administrative guidance to ensure that the Dissemination Application has been installed and started in a secure manner, as intended by the developer. D.4.1 The administrative guidance shall cover configuration, maintenance, and administration of the Dissemination Application in a secure manner. D.4.2 The administrative guidance shall describe the functions and interfaces available to the administrator in addition to how to manage the Dissemination Application in a secure manner. D.4.3 The administrative guidance shall describe all security requirements for the Dissemination System that are relevant to the administrator and the Dissemination Application.
O.AUDIT_GENERATION: The Dissemination System will provide the capability to detect and create records of security-relevant events associated with users.
A.1.1 The Dissemination System shall have configurable auditing capabilities. Audit levels shall be hierarchical from the least amount of information to the most. The Dissemination System shall support the following audit levels: alert, critical, error, warning, notice, information, and debugging. All audited events shall be recorded. A.1.2 The date and time of the event, number of bytes sent to the server, the remote host name or IP address, type of event, user identity (if applicable), and the outcome (success or failure) shall be recorded. C.1.1 The Dissemination Application shall have configurable auditing capabilities. All audited events

shall be recorded. C.1.2 The date and time of the event, type of event, user identity (if applicable), and the outcome (success or failure) shall be recorded.
O.AUDIT_PROTECTION: The Dissemination System will provide the capability to protect audit information.
A.1.4 An authorized administrator shall be able to select the amount of time between audit log rotations. This shall be configurable for daily, weekly, or monthly rotations. Additionally, the administrator shall be able to specify how many rotated logs are kept on the system before being archived.
O.AUDIT_REVIEW: The Dissemination System will provide the capability to view audit information.
A.1.3 The audit records generated by the Dissemination System shall be in a format that can be parsed. C.1.3 The audit records generated by the Dissemination Application shall be in a format that can be parsed.
O.CONGIFURATION_MANAGEMENT: The configuration of and all changes to the Dissemination System will be tracked and controlled throughout the Dissemination System's development.
B.1.1 The Dissemination System's third party software and documentation including configuration files shall be maintained under Configuration Management. D.1.1 The Dissemination Application documents (e.g. functional specification) and software shall be archived using the CM process. D.5.1 The Dissemination Application shall follow the same spiral life cycle model and procedures as the TCX project.
O.DISPLAY_BANNER: The Dissemination System will display an advisory warning regarding use of the Dissemination System.
A.6.1 The Dissemination System shall clearly display an access banner describing the restrictions of use, legal agreements, or any other appropriate information to which users consent by accessing the system. B.2.2 The user guidance shall clearly present all user responsibilities necessary for secure use of the Dissemination System, including those related to assumptions regarding user behavior found in the access banner.
O.DOCUMENTED_DESIGN: The design of the Dissemination Application will be adequately and accurately documented.
D.3.1 An informal high level design specification shall be developed for the Dissemination Application. D.3.2 The design of the Dissemination Application shall meet the functional requirements.

O.PARTIAL_FUNCTIONAL_TESTING: The Dissemination System will undergo some security functional testing that demonstrates that it satisfies some of its security functional requirements.
B.3.1 The Dissemination System test plan shall cover expected usage of the Dissemination System in addition to limited testing of unexpected situations. This partial functional testing shall ensure that the Dissemination System properly performs functions required for dissemination. D.6.1 The developer shall develop a test plan to cover expected usage of the Dissemination Application in addition to limited testing of unexpected situations. This partial functional testing shall ensure that the Dissemination Application properly handles access to releasable items while maintaining its own configuration.
O.SECURE_DELIVERY: The Dissemination System will provide mechanisms for users to verify that any data disseminated matches the master version maintained by the Dissemination System.
C.2.3 The Dissemination Application shall not modify the digital signatures placed on the dissemination material by the Configuration Management system.
O.SOUND_IMPLEMENTION: The implementation of the Dissemination Application will be an accurate instantiation of its design.
D.3.3 The Dissemination Application shall be implemented in accordance with the design.
O.SYSTEM_ACCESS: The Dissemination System will provide mechanisms that control a user's logical access to it.
A.5.1 The Dissemination System shall ensure that users are identified and authenticated in order to associate them with the proper security attributes while accessing data. Security attributes shall include but are not limited to the user's identity and the group(s) to which that user belongs. A.5.2 The Dissemination System shall authenticate registered users based on their user ID and password. A.5.3 The Dissemination System shall authenticate users prior to allowing access to any non-public documents on the Dissemination System.
O.TIME_STAMPS: The Dissemination System will use time stamps for accountability purposes.
A.1.2 The date and time of the event, number of bytes sent to the server, the remote host name or IP address, type of event, user identity (if applicable), and the outcome (success or failure) shall be recorded. C.1.2 The date and time of the event, type of event, user identity (if applicable), and the outcome (success or failure) shall be recorded.
O.USER_CONFIDENCE: The Dissemination System will provide mechanisms that permit end users to have confidence that received controlled-access data comes from the Dissemination System.

A.2.1 The Dissemination System shall employ cryptographic functionality to provide a secure connection between the Dissemination System and the users. A.3.1 The Dissemination System shall use a digital certificate signed by an authorized Certificate Authority for authenticating itself to the users.
O.USER_GUIDANCE: The Dissemination System will provide users with the necessary information for secure data access.
B.2.1 The user guidance shall describe the interaction between the user and the Dissemination System for proper retrieval of releasable data. B.2.2 The user guidance shall clearly present all user responsibilities necessary for secure use of the Dissemination System, including those related to assumptions regarding user behavior found in the access banner.
O.VULNERABILITY_ANALYSIS: The Dissemination Application will undergo some vulnerability analysis to demonstrate the design and implementation do not contain any obvious flaws.
D.6.2 To help mitigate misuse of the Dissemination Application the guidance documentation shall be complete, clear, consistent, and reasonable. It shall list the assumptions about the environment, and requirements for external security measures. D.6.3 The developer shall perform a vulnerability assessment of the Dissemination Application along with provision of vulnerability analysis documentation.
PART II: ENVIRONMENT REQUIREMENTS MAPPING
OE.DATA_MARKING: All dissemination data will be properly marked with access control attributes by the Dissemination Environment.
F.2.1 The Dissemination Environment shall properly mark all dissemination data with access control attributes.
OE.INTERNAL_PROCESSING: The internal implementation and execution of the Dissemination System's underlying operating system, web server, and cryptographic libraries will run as expected.
This objective addresses the assumption A.INTERNAL_PROCESSING and has no corresponding security functional requirement.
OE.MANAGE: The IT Environment will provide all the functions and facilities necessary to support the administrators in their management of the security of the Dissemination System, and restrict these functions and facilities from unauthorized use.

E.1.1 The IT Environment shall restrict the ability to perform the following functions to the authorized administrator: enable, disable, and modify the audit settings; backup and restore audit records; adjust the configuration parameters; and modification of any databases used by the Dissemination Application. E.1.2 The IT Environment shall restrict the ability to modify the security attributes of both data and users of the system to the authorized administrator. E.2.1 The IT Environment shall lock a local interactive administrator session after a specified period of inactivity by disabling system access to data and display devices other than the ability to unlock the session. The IT Environment shall require re-authentication by the administrative user prior to unlocking the session. E.5.1 The IT Environment shall ensure that users are identified and authenticated in order to associate them with the proper security attributes while accessing data. Security attributes shall include but are not limited to the user's identity and the group(s) to which that user belongs.
OE.NO_REMOTE_ADMIN: The IT Environment will provide only local capabilities for Dissemination System administration.
This objective addresses the assumption A.NO_REMOTE_ADMIN and has no corresponding security functional requirement.
OE.NO_ROGUE_ADMIN: The Dissemination Environment shall ensure that administrators are non-hostile, appropriately trained, and follow all administrator guidance.
This objective addresses the assumption A.NO_ROGUE_ADMIN and has no corresponding security functional requirement.
OE.NO_ROGUE_ENTITY: The Dissemination Environment shall ensure that all of its entities are trustworthy and protect both the dissemination data and the data they generate to support secure distribution.
F.3.1 The Dissemination Environment shall ensure that all data generated by its entities required for proper dissemination is protected in situ and during transit.
OE.PHYSICAL: Physical security, commensurate with the value of the Dissemination System and the data it contains, will be provided by the Dissemination Environment.
This objective addresses the assumption A.PHYSICAL and has no corresponding security functional requirement.
OE.REGISTERED_USERS: The Dissemination Environment will provide a mechanism for users to register prior to accessing non-public material on the Dissemination System.
F.1.1 The Dissemination Environment shall provide a mechanism for users to register prior to accessing

non-public material on the Dissemination System.
OE.RELIABLE_TIME_STAMP: The IT Environment will provide reliable time stamps.
E.4.2 The IT Environment shall be able to provide time stamps for its own use.
OE.SYSTEM_PROTECTION: The IT Environment will provide sufficient mechanisms to protect the Dissemination System's data and memory during storage and execution.
E.3.1 The IT Environment shall restrict the ability to create or modify webpage content to authorized administrators. E.3.2 The IT Environment shall be capable of limiting the ability to create or modify server executable content. E.3.3 The IT Environment shall protect the Dissemination System's private key from unauthorized modification and viewing. E.4.1 The IT Environment shall restrict all non-administrative users of the Dissemination System from reading from and writing to the audit trail.

Table 8. Requirements Mapping

H. SUMMARY

The functional and assurance requirements for the Dissemination System and the Dissemination Application were specified. Functional security requirements for the Dissemination Environment and the IT Environment were also presented. The mapping of the assumptions, threats, policies, security objectives, and requirements was then produced. Additionally, the rationale for some of the mapping was presented. The next chapter covers the top level design of the initial implementation and the complete design specification.

THIS PAGE INTENTIONALLY LEFT BLANK

V. DESIGN SPECIFICATION

A. INTRODUCTION

The functional security requirements described in Chapter IV provide the basis for the design of the initial implementation of the Dissemination System. This chapter includes a design overview and a high level design specification. The design overview briefly discusses the system processes and databases used by the Dissemination System. Following the overview is the complete design specification.

B. DESIGN OVERVIEW

This section provides a brief description of the design of the initial implementation of the Dissemination System.

The Dissemination System is made up of multiple system processes and a set of supporting tools running on an open-source operating system. The system processes are an Apache Web Server, the TCX Dissemination Application, *crond, logrotate,* and *webalizer.* The last three processes are collectively known as administrative tools. The supporting tools include the OpenSSL tool and the *linkcheck.pl* Perl script. The figure below depicts the architecture of the Dissemination System.

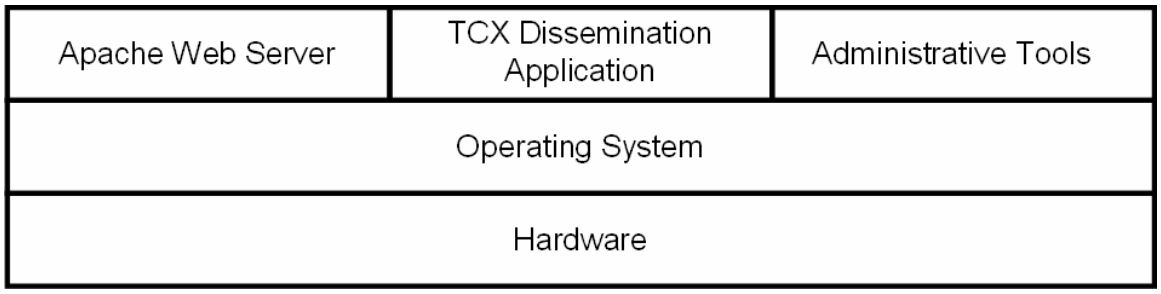


Figure 3. Dissemination System Architecture

Each system process is responsible for implementing specific services. The web server performs password-based user authentication, group-based access control, configurable web server audit logging, web hosting services and SSL protected connections. The Dissemination Application performs redaction based on the Dissemination Policy database and the Releasable Items List database; removal of revoked or non-releasable documents, i.e. sweeping; webpage management; and application specific audit logging. The *crond* process oversees the execution of the other

administrative tools. The *logrotate* daemon is responsible for rotating the audit logs for archival purposes. The *webalizer* daemon provides the Apache access log in an HTML format for viewing and analysis. The table below summarizes the services implemented by each component.

Web Server	• Access Control • Audit Logging • Authentication • SSL Protected Connections • Web Hosting
Dissemination Application	• Audit Logging • Redaction • Sweeping • Web Page Management
Administrative Tools	• Analyze Audit Logs • Rotate Audit Logs • Run daemons

Table 9. Dissemination System Services Implemented

The Dissemination System maintains eight databases. Each database is utilized by one or more of the system processes. The User database contains authentication data for registered users of the Dissemination System. The Releasable Items List database specifies the list of releasable material. The Dissemination Policy database specifies how the XML tags are used to control the dissemination of releasable material. The Dissemination System Key database contains the private/public key pair. The keys are 1024-bit RSA keys stored in PEM format. The Dissemination System Certificate database contains the X.509 certificate of the Dissemination System that is signed by the CISR CA server and stored in PEM format. The Dissemination Material Repository database contains the TCX project material from Configuration Management and the generated HTML views. The Webpage Repository database contains the symbolic links to releasable dissemination material and the TCX project and user group homepages. The Audit Log Repository database contains the access and error logs generated by the Apache Web Server and the Dissemination Application. The databases are either static

(i.e. created during system setup or imported from an external source) or dynamic (i.e. created or modified during runtime). Only read access is allowed for static databases. Only processes that need to update a database will have write access to that database. The databases and their access modes are summarized in Table 10.

Database	Configuration	Accessed By	Access Mode
User	Static	Web Server	Read
Releasable Items List	Static	DA	Read
Dissemination Policy	Static	DA	Read
DS Key	Static	Web Server	Read
DS Certificate	Static	Web Server	Read
Dissemination Material Repository	Dynamic	DA	Read/Write
Webpage Repository	Dynamic	DA	Read/Write
		Web Server	Read
Audit Log Repository	Dynamic	DA, Web Server	Read/Write
		Tools	Read/Write

Table 10. Database Mapping

The system requirements, detailed database descriptions, and process flows are described in the high level design specification in the next section.

C. HIGH LEVEL FUNCTIONAL DESIGN SPECIFICATION

1. Introduction

The Dissemination System will be implemented on a server platform running a Linux-based operating system that is capable of providing classic UNIX access control with elevated administrator privileges.

An Apache web server will be used to host the Dissemination System web site. The web server will implement the following functionality: group-based access control to TCX project material, user authentication, web server auditing and management of SSL

protected connections between the web server and client machines. Apache will require the client browser to establish SSL connections for access to non-public web content.

The Dissemination Application is a system process that is responsible for preparing the dissemination material for online distribution. It performs the following operations: sweeping the dissemination material repository, redaction, webpage management, and application audit.

These three key components (OS, web server, and DA) form the core of the Dissemination System and allow the secure dissemination of the TCX project material. Additionally there will be administrative tools and supporting tools on the system which will be used to maintain the Dissemination System.

2. Requirements

2.1 Dissemination System Requirements

All of the functional requirements for the Dissemination System, the Dissemination Application, the IT Environment and the Dissemination Environment are defined in Chapter IV. The assurance requirements for the Dissemination System and Dissemination Application are also included. The functional requirements are collectively fulfilled by the IT Environment, the Apache Web Server, the Dissemination Application, the Administrative Tools, the Supporting Tools, and the Dissemination Environment. The following table maps the functional requirements from Chapter IV to the system components that implement them.

A.1 Dissemination System Audit	
A.1.1	Apache Web Server
A.1.2	Apache Web Server
A.1.3	Apache Web Server
A.1.4	Administrative Tool (*logrotate*)
A.2 Dissemination System Communication	
A.2.1	Apache Web Server

A.3 Dissemination System Cryptography	
A.3.1	Apache Web Server
A.4 Dissemination System User Data Protection	
A.4.1	Apache Web Server
A.5 Dissemination System Identification and Authentication	
A.5.1	Apache Web Server
A.5.2	Apache Web Server
A.5.3	Apache Web Server
A.6 Dissemination Application Access	
A.6.1	Apache Web Server
C.1 Dissemination Application Audit	
C.1.1	Audit Handler
C.1.2	Audit Handler
C.1.3	Audit Handler
C.2 Dissemination Application User Data Protection	
C.2.1	Dissemination Application (Redaction Function)
C.2.2	Dissemination Application (Sweeping Function)
C.2.3	Dissemination Application (Webpage Management Function)
C.2.4	Dissemination Application (Redaction Function)
E.1 IT Environment Security Management	
E.1.1	IT Environment
E.1.2	IT Environment
E.2 IT Environment Access	

E.2.1	IT Environment
E.3 IT Environment Data Protection	
E.3.1	IT Environment
E.3.2	IT Environment
E.3.3	IT Environment
E.4 IT Environment Audit	
E.4.1	IT Environment
E.4.2	IT Environment
E.5 IT Environment Identification and Authentication	
E.5.1	IT Environment
F.1 Dissemination Environment Security Management	
F.1.1	Dissemination Environment
F.2 Dissemination Environment User Data Protection	
F.2.1	Dissemination Environment
F.3 Dissemination Environment Protection	
F.3.1	Dissemination Environment

Table 11. Functional Requirements Mapping

The assurance requirements are collectively met by DS/DA Configuration Management, User Guidance documentation, User Access Banners, Administrative Guidance documentation, the DS/DA Design Specification, DA Development Specification, DA Implementation Representation, DA Vulnerability Assessment Report, and DA Life Cycle Management. The mapping of the assurance requirements to these elements is shown in the following table.

B.1 Dissemination System Configuration Guidance	
B.1.1	DS/DA Configuration Management

B.2 Dissemination System Guidance Documents	
B.2.1	User Guidance
B.2.2	User Guidance & User Access Banners
B.2.3	Administrative Guidance
B.2.4	Administrative Guidance
B.2.5	Administrative Guidance
B.2.6	Administrative Guidance
B.3 Dissemination System Testing	
B.3.1	DS/DA Design Specification
D.1 Dissemination Application Configuration Management	
D.1.1	DS/DA Configuration Management
D.2 Dissemination Application Operation	
D.2.1	Administrative Guidance
D.3 Dissemination Application Development	
D.3.1	DS/DA Design Specification
D.3.2	DS/DA Design Specification
D.3.3	DA Development Specification, DA Implementation Representation
D.4 Dissemination Application Guidance Documents	
D.4.1	Administrative Guidance
D.4.2	Administrative Guidance
D.4.3	Administrative Guidance
D.5 Dissemination Application Life Cycle Support	
D.5.1	DA Life Cycle Management

D.6 Dissemination Application Testing and Vulnerability Assessment	
D.6.1	DS/DA Design Specification
D.6.2	User Guidance & Administrative Guidance
D.6.3	DA Vulnerability Assessment Report

Table 12. Assurance Requirements Mapping

This design specification only addresses the functional requirements of the Dissemination System and Dissemination Application. Assurance requirements will be addressed as future work.

2.2 User Requirements

All users of the Dissemination System must accept the terms and conditions of the Dissemination System web site which are promulgated on web page banners. Access to restricted web pages additionally requires that the user must register with the CISR web server to obtain a valid username and password and that the user must use a web browser configured for SSL with the CISR CA digital certificate installed. All of these procedures are described in the user guidance documentation.

3. Databases

The databases are kept in different locations in the file system. Some of these database locations are predefined by the operating system. These include the DS Key database, the DS Certificate database, and the Audit Log Repository database. Other databases are located in the DS Root, Document Root, or Web Root directories. This file structure is illustrated in Figure 4, which also shows the location of the configuration files for the system processes and their output files.

The definitions of the databases include the terms white space and carriage return. For this specification white space is an ASCII blank space (a hexadecimal value of 20). A carriage return is the UNIX carriage return which consists of an ASCII line feed (a hexadecimal value of 0A). Aside from these characters, only printable characters will be used.

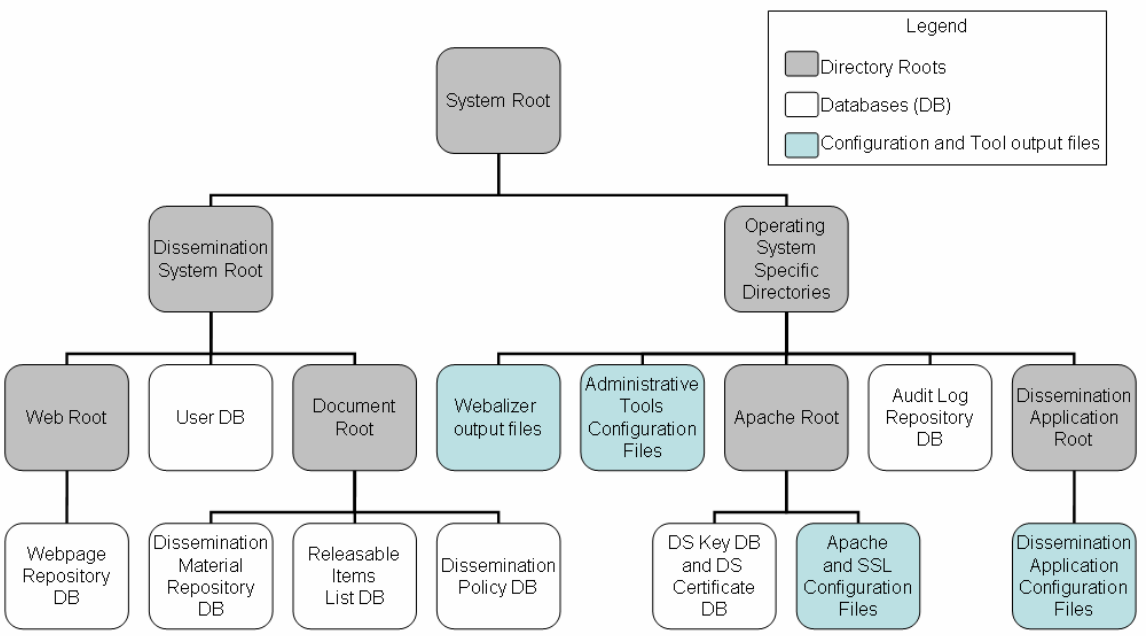


Figure 4. Directory Structure

3.1 User Database

The User database contains authentication data used to identify and authenticate the user. The database is a read-only input database. This distributed database consists of two read-only ASCII text files: a Password file and a Group file. The Password file contains a list of entries, with each entry terminated by a carriage return. Each entry contains the following fields:

- Username – the unique 8-character username of a register user of the system
- Password hash – a 16-byte MD5 checksum of the user password generated using an Apache compatible password hash generation tool

The two fields in the Password file are separated by either a colon or a colon and at least one white space.

The Group file consists of a list of entries with each entry consisting of the following fields:

- Group Name – a valid group name on the system

- Group Members – a variable length list containing the names of users belonging to the group specified in Group Name. The user names are separated by at least one white space.

For each entry, the Group Name and Group Members fields are separated by a colon and at least one white space. Each entry is terminated by a carriage return.

The User database will be imported from the CISR web server and used by the Apache web server for access control to different portions of the webpage.

3.2 Releasable Items List Database

The Releasable Items List (RIL) database contains a list of all Configuration Management items that are releasable at a given time. The RIL will be used in conjunction with the Dissemination Policy database for distribution of project material in accordance with the TCX dissemination policy. It is a read-only input database. This database is an ASCII text file containing a list of entries, with each entry terminated by a carriage return. Each entry contains the following fields:

- Filename – the full path and filename of the releasable item
- DDF Filename – the name of the corresponding document descriptor file (DDF)

The two fields must be separated by at least one white space. The Dissemination Application uses this database during sweeping, redaction, and webpage management. For the initial implementation, the document descriptor file specifies the file-level access control markings for each file in a Configuration Item.

3.3 Dissemination Policy Database

The Dissemination Policy database specifies how the TCX access control markings (implemented as XML tags in the DDF) are used to control the dissemination of releasable material. It is a read-only input database. This database is an ASCII text file containing a list of entries, with each entry terminated by a carriage return. Each entry contains the following fields separated by at least one white space:

- File Type Marking– a printable 32-character ASCII string specifying the type of the dissemination material
- File Sensitivity Marking – a printable 32-character ASCII string specifying the distribution sensitivity of the dissemination material. This marking is optional and the character * (an asterisk) specifies that the field is not in use
- Authorized User Groups – a variable length list specifying the user groups that are authorized to view material that is marked with the corresponding combination of File Type and File Sensitivity markings. The group names are separated by a comma and at least one white space.

The access control markings currently defined for the initial implementation are listed in Table 13 and Table 14.

File Type Marking
Code
Engineering Notes
Publications
Specifications
User Documents
Verification Evidence

Table 13. File Type Markings

File Sensitivity Marking
Engineering Release
Proprietary Restricted

Table 14. File Sensitivity Markings

All releasable items must be labeled with a File Type marking. Only releasable items that require special dissemination handling will have a File Sensitivity marking.

The Dissemination Policy database is utilized by the Dissemination Application to create redacted views and to update the webpage repository.

3.4 Dissemination System Key Database

The Dissemination System Key database contains the private/public key pair. They are generated locally, i.e., on the Dissemination System, using the OpenSSL tool. The keys are 1024-bit RSA keys stored in PEM format in two separate files. Following the Apache convention, the key files are kept together with the Apache configuration files. The private key is encrypted and requires a pass phrase to decrypt before use. The public key is kept as part of the certificate signing request file that is generated along with the key pair. Only the Dissemination System administrator has write access to this database. The keys are used by the Apache web server to establish and maintain SSL connections with the clients.

3.5 Dissemination System Certificate Database

The Dissemination System Certificate database contains the X.509 certificate of the Dissemination System that is signed by the CISR CA server and stored in PEM format. The certificate file is located inside the Apache configuration files folder. The certificate is used by the web server to authenticate itself to the client machine when initiating a SSL connection.

3.6 Dissemination Material Repository Database

The Dissemination Material Repository database contains the TCX project material from Configuration Management and the redacted views. The Dissemination Material Repository is a distributed database made up of the following components:

- XML documents – all XML documents to be disseminated
- Non-XML documents – all non-XML documents to be disseminated
- Document descriptor files – a set of files containing XML access control markings for all dissemination material
- Redacted documents – all redacted XML files and their corresponding HTML generated views

- Revoked documents – all documents that are swept from the other components of the Dissemination Material Repository database

The XML documents, non-XML documents, and document descriptor files are read only files. They are imported from the Releasing Agent and processed by the Dissemination Application to create the Webpage Repository. The Redacted documents component is a dynamic portion of the Dissemination Material Repository database that is generated and maintained by the Dissemination Application.

3.7 Webpage Repository Database

The Webpage Repository database contains all web viewable content. The Webpage Repository database contains two types of data:

- Symbolic links – refer to files in the Dissemination Material Repository database
- Homepages – project and group top level web pages

The symbolic links are generated and maintained by the Dissemination Application. Symbolic links are used instead of the target files for maintenance and security reasons. On the Dissemination System there can exist multiple symbolic links to the same target file located in the Dissemination Material Repository database. Based on the target file's access control markings and the current dissemination policy, the target file can be released to multiple user groups. Hence multiple links must be created in the appropriate group directories in the Webpage Repository database. System maintenance becomes easier because content changes to the target file require no changes to the associated symbolic links. The symbolic links provide additional security by disallowing web users from accessing the target files directly.

Most of the homepage files contain content that is dynamically generated by the Dissemination Application; however, from the user's viewpoint the pages are static. All users of the Dissemination System have read-only access to different parts of the webpage repository based on their group authorization. The entire Webpage Repository database is hosted by the Apache web server.

3.8 Audit Log Repository

The Audit Log Repository database contains the web server and Dissemination Application audit logs. The repository contains two types of logs:

- Error Log – contains error messages generated by either the web server or the Dissemination Application
- Access Log – contains informative messages about accesses to material on the web server or operations performed by the Dissemination Application that require auditing

The Audit Log Repository is regularly updated by the web server and the Dissemination Application. The audit logs contain all audit material from the Dissemination System based on the current audit policy as specified by the Program Manager.

4. Processes

4.1 Apache Web Server

The Apache Web Server is responsible for user authentication, access control, audit logging, web hosting, and management of SSL protected connections.

4.1.1 Input

The Apache Web Server requires the following input databases:

- Webpage Repository database – used for web hosting
- User database – used for authentication and access control
- DS Key database – used during process startup
- DS Certificate database – used for SSL protected connections

4.1.2 Output

The Apache Web Server logs auditable events to both the Access and Error logs contained within the Audit Log Repository database. This is done by the audit logging functionality of the web server:

- Audit Log Repository database – used for audit logging

4.1.3 Processing

4.1.3.1 Initialization

Prior to running the web server, the administrator must modify the default Apache configuration file to support TCX online distribution. Specifically, the administrator must set the web root directory path, specify access controls for web content, and enable SSL. The web root directory is used to specify the Webpage Repository database used for web hosting. Apache is configured to use the Basic Authentication method to authenticate users based on the Password file of the User database. Apache is also configured to use the Group file of the User database to determine group membership of users and the directory access control attributes to enforce group-based access controls. In order to provide SSL protected communications, Apache must be configured with the OpenSSL module installed. This allows Apache to utilize the OpenSSL library to handle HTTPS requests. Additionally, Apache rewrite rules are used in the Apache configuration file to force the use of SSL accesses to non-public proprietary directories that require authentication.

4.1.3.2 Runtime

During runtime the Apache Web Server is responsible for serving web pages to users and protecting non-public material. For public users the web server acts as a typical web server, serving user requests and allowing the user to access non-proprietary material. For registered users the web server additionally establishes SSL connections, performs user authentication and access control, and audits user accesses.

A typical registered user web request is handled as follows:

1. The web server responds to the user request and provides the project homepage from the Webpage Repository database.

2. If the user selects a proprietary material link, the web server establishes an SSL connection with the client machine and initiates the user authentication sequence. Upon receiving user authentication data (username and password) the web server verifies the authentication data from the Password file of the User database.

3. After successful authentication, the web server uses the group-based access control information in its configuration file and the group membership data in the Group file of the User database to determine if the user is authorized to access the data.

4. The user is then provided with the group homepage containing links to all material which is accessible by that particular group.

5. User accesses are logged, in addition to web server errors encountered during the handling of the user request. The level of detail captured in the audit logs is dependent on the audit level specified in the Apache configuration file.

A SSL connection must be established prior to authentication in order to provide confidentiality protection for the transmission of authentication data and non-public dissemination material. When requested by the client, the web server presents its digital certificate stored in the Dissemination System Certificate database to identify itself. The client is not required to authenticate itself to the web server.

4.2 Dissemination Application

The Dissemination Application is a system process responsible for sweeping, redaction, webpage management, and audit. These functions are implemented in the following modules: Sweeping Handler, Redaction Handler, Webpage Manager, and Audit Handler.

4.2.1 Input

The Dissemination Application requires the following read-only databases:

- Releasable Items List database – used by Sweeping Handler, Redaction Handler, and Webpage Manager
- Dissemination Policy database – used by Redaction Handler and Webpage Manager
- Dissemination Material Repository database – used by Sweeping Handler, Redaction Handler, and Webpage Manager

4.2.2 Output

The Dissemination Application generates or modifies the following databases:

- Dissemination Material Repository database – modified by Sweeping Handler and Redaction Handler
- Webpage Repository database – generated or modified by Webpage Manager
- Audit Log Repository database – generated or modified by Sweeping Handler and Audit Handler

4.2.3 Processing

4.2.3.1 Initialization

The Dissemination Application is a program that runs in the background as a daemon. The Dissemination Application configuration file specifies the location of all input and output databases used by the application. The Dissemination Application is executed by the *cron* daemon (*crond*) on a regular basis. The frequency of execution is specified in the *crontab* file.

4.2.3.2 Runtime

4.2.3.2.1 Sweeping Handler

The Sweeping Handler assures that all items in the XML documents and Non-XML documents components of the Dissemination Material Repository database are specified on the Releasable Items List database. For items in those components that are not on the RIL the Sweeping Handler moves them to the Revoked components of the database. Revocation of files on the Dissemination System is automatically performed when a new RIL (without the revoked item on it) is imported and the Dissemination Application is run. The second function of the sweeping module scans the document root directory and the web root directory to locate broken symbolic links. If broken links are found, their location is recorded in the Dissemination Application Error Log. An alarm will be generated to notify the system administrator if so configured.

4.2.3.2.2 Redaction Handler

The primary function of the Redaction Handler is to generate proper HTML views of releasable XML documents based on the access control markings and the current

dissemination policy. For the initial implementation access control markings are applied at the file level. The Redaction Handler performs the following steps for each XML file specified in the RIL:

1. From the DDF file of the XML file, the Redaction Handler locates and extracts the access control tags for the XML document.

2. The Dissemination Policy database is examined to determine the Authorized User Groups for the specified pair of access control tags.

3. The Redaction Handler creates a folder with the same name as the XML filename in the Redacted Views component of the Dissemination Material Repository database.

4. The Redaction Handler then creates redacted XML file(s) in that folder based on the XML file and the Authorized User Groups. A redacted XML file will be created for each authorized user group. This file will be named as follows: <original name><_><user group name><.xml>, e.g. specification_developers.xml.

5. Each redacted XML file is then processed through XSL transformations to create an HTML generated view with an HTML link to the XML file appended.

4.2.3.2.3 Webpage Manager

The Webpage Manager is responsible for managing the Webpage Repository database. Specifically, it creates symbolic links in the Webpage Repository database for the HTML generated views and non-XML files, and updates the group homepages to reflect the latest group-viewable content. The Webpage Repository database contains a project home folder and a subfolder for each user group. The symbolic links created by the Webpage Manager are stored in the group specific folders. The HTML generated views and non-XML files are located in the Dissemination Material Repository database. The HTML generated views are created from the redacted XML documents by the Redaction Handler as described above.

The Webpage Manager performs the following steps for each redacted XML document located in the Redacted Views component of the Dissemination Material Repository database:

1. The Webpage Manager extracts the user group name from the filename of the redacted XML document. The format of the filename is described in the Redaction Handler section above.

2. The Webpage Manager creates a symbolic link to the HTML generated views associated with the redacted XML document in the group specific folder of the Webpage Repository database. The symbolic link has the same name as the target file (i.e., the HTML generated view).

The creation of symbolic links for non-XML files is a more extensive process. The Webpage Manager performs the following steps for each non-XML file specified in the Releasable Items List database:

1. From the DDF file of the non-XML file, the Webpage Manager locates and extracts the access control tags for the file.

2. The Dissemination Policy database is examined to determine the Authorized User Groups for the specified pair of access control tags.

3. The Webpage Manager creates a symbolic link in the group specific folder of the Webpage Repository database with the target being the non-XML file. The symbolic link has the same name as the target file located in the non-XML documents component of the Dissemination Material Repository database.

The second function of the Webpage Manager is to update the group homepages. The Homepages component of the Webpage Repository database contains both static and dynamic elements. The TCX project homepage is statically created by the system administrator to display links to the different group sections of the TCX Dissemination System website. The group homepages are dynamically generated by the Webpage Manager based on the directory content.

4.2.3.2.4 Audit Handler

The Audit Handler is responsible for logging all actions performed by the Dissemination Application. Audit logs will be generated in the Common Log Format as specified in the Dissemination Application configuration file. Audit logs are generated and stored in the Audit Log Repository database.

4.3 Administrative Tools

4.3.1 *crond*

crond is a system daemon that executes scheduled commands. To do this, it scans *crontab* files, which specify daemons to be run, and runs their commands at the appropriate time [13]. For the Dissemination System the *crontab* files will specify the *logrotate*, *webalizer* and the Dissemination Application processes to be run on a system administrator specified interval.

4.3.2 *logrotate*

logrotate rotates, compresses, and mails system logs on a set interval [14]. The daemon allows the administrator to specify how frequently the logs are rotated (daily, weekly, or monthly) and how many old rotations to keep on the system. *logrotate* appends a number to the current log name and creates a new blank current log. If older rotations exist, then those are renamed such that the oldest log name will contain the highest number. For the Dissemination System, *logrotate* is responsible for rotating the Apache log files utilized for auditing of the web server and the log files utilized by the Dissemination Application. The Dissemination System administrator is responsible for configuring *logrotate* in accordance with the specification set forth by the TCX project Manager.

4.3.3 *webalizer*

webalizer is a web server log file analysis tool [15]. *webalizer* creates HTML graphical representations of the Apache access log. The HTML page contains statistics and accesses to web page content recorded in the access log. For the Dissemination System, *webalizer* will be used solely by the system administrator to analyze the access audit log for non-proprietary information access on the web server. Since it runs as a daemon the *webalizer* output will be updated to reflect the most current access log. If desired, the system administrator can run *webalizer* to obtain the timeliest update.

5. Supporting Tools

5.1 OpenSSL

The OpenSSL tool provides commands for generating public/private key pairs and certificates. For the Dissemination System the OpenSSL tool will be used to generate an RSA public/private key pair for the web server and an X.509 certificate signing request. The public/private key pair is stored in the Dissemination System Key database. The signing request is sent to the CISR CA and the returned signed certificate is stored in the Dissemination System Certificate database.

5.2 *linkcheck.pl*

The *linkcheck.pl* Perl script locates broken symbolic links on a Linux-based operating system. The Dissemination System utilizes this script to assure that the dissemination material repository and the webpage repository do not contain broken links to target files.

D. SUMMARY

An overview of the design specification was first presented. Following the overview, a high level design specification containing the details of the databases, processes, and procedures required to implement the Dissemination System was provided. The initial implementation of the Dissemination System based on this specification is discussed in the next chapter.

THIS PAGE INTENTIONALLY LEFT BLANK

VI. INITIAL IMPLEMENTATION

The TCX Dissemination System initial design was discussed in the previous chapter. This chapter describes the construction of the prototype of this design starting with the development and testing environment. Next, the process used to setup an operational prototype system is discussed and includes a description of the manual simulation of the Dissemination Application. A set of test scenarios is presented along with their result. This prototype partially implements the design; the unimplemented features are documented. The final section describes the problems encountered during development and their solutions.

A. DEVELOPMENT AND TESTING ENVIRONMENT

The prototype Dissemination System (herein referred to as prototype system) was implemented on a desktop machine running Fedora Core 3 Linux. The Apache HTTP Server 2.0 software was used as the web server for the system. For developmental testing, the prototype system was configured to respond to the default local computer address, i.e., *localhost*.

For system testing, the prototype system was connected to the NPS campus network and assigned an NPS domain IP address. With this address, other computers within the NPS network were able to connect to the prototype system. The prototype system was disconnected from the NPS network except for selected remote access test scenarios.

Both Linux and Windows client systems were used to conduct the testing. The Linux client test machine was the same system hosting the prototype system. The Windows client test system was a laptop attached to the NPS network. Testing was done using two different operating systems with two different web browsers (Mozilla and Internet Explorer) to assure that the access control methods implemented were compatible with both systems. Figure 5 illustrates the test topology.

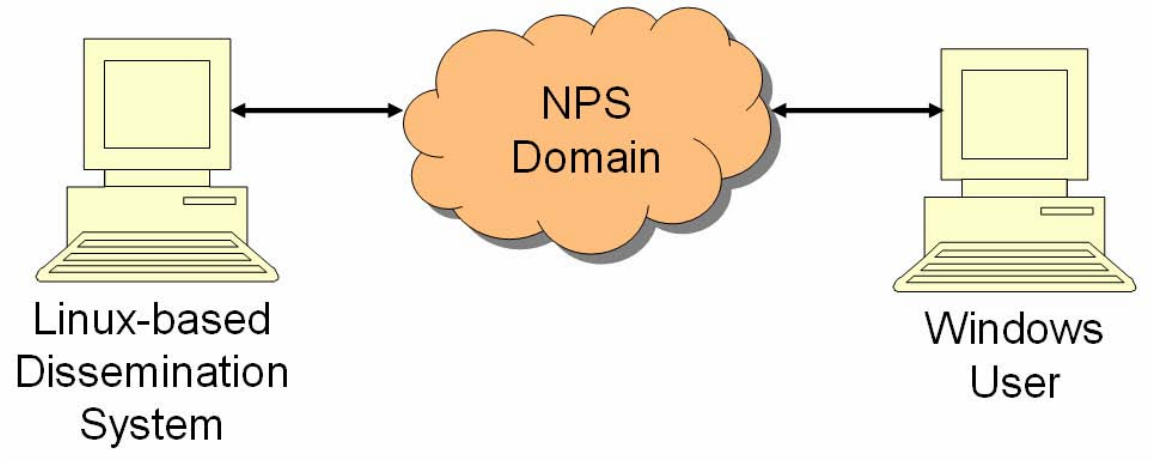


Figure 5. Testing Environment

B. SYSTEM CONSTRUCTION

The construction of the prototype system consisted of four stages:

1. File System Customization
2. Database Generation
3. Manual Simulation of the Dissemination Application
4. Web Server Configuration

The details of each stage are described separately below.

1. File System Customization

The directory structure of the Dissemination System must be implemented prior to the generation of databases. Some of the databases are distributed and thus require the creation of multiple directories. Other databases use default operating system folders and therefore will not need to have new directories created. Table 15 describes the directory structure, the files contained in each directory, and the mapping of the files to the databases.

Path	What it contains	Database
/var/www/	DS Root Folder	n/a
/etc/httpd/	Apache Root Folder	n/a
/var/www/html/	Web Root Folder	n/a

/var/www/documents/	DS Document Root	n/a
/etc/httpd/conf/	Apache Web Server configuration file	n/a
/etc/httpd/conf.d/	SSL, *webalizer*, and *logrotate* configuration files	n/a
/etc/httpd/conf.d/ssl.key/	private key	DS Key
/etc/httpd/conf.d/ssl.crt/	digital certificate	DS Certificate
/var/www/usage/	webalizer output files	n/a
/var/logs/	error and access logs	Audit Log Repository
/var/www/login/	.htpasswd & .groups files	User
/var/www/documents/ddf/	document descriptor files	Dissemination Material Repository
/var/www/documents/non-XML_original/	Non-XML dissemination material	Dissemination Material Repository
/var/www/documents/revoked/	Files that are swept from the DS by the DA	Dissemination Material Repository
/var/www/documents/XML_original/	XML dissemination material	Dissemination Material Repository
/var/www/documents/html_generated_views/	HTML generated views of XML	Dissemination Material

	documents and links to XML originals	Repository
/var/www/documents/ril	Releasable Items List	Releasable Items List
/var/www/documents/policy	Current release policy file	Dissemination Policy
/var/www/html/tcx/	Root homepage (any files in the tcx folder, including subfolders are web accessible)	Webpage Repository
/var/www/html/tcx/administrator/	Administrator accessible content	Webpage Repository
/var/www/html/tcx/collaborator/	Collaborator accessible content	Webpage Repository
/var/www/html/tcx/consumer/	Consumer accessible content	Webpage Repository
/var/www/html/tcx/developer/	Developer accessible content	Webpage Repository
/var/www/html/tcx/evaluator/	Evaluator accessible content	Webpage Repository
/var/www/html/tcx/nist_nsa/	NIST/NSA accessible content	Webpage Repository
/var/www/html/tcx/public/	Publicly accessible content	Webpage Repository

Table 15. Directory Structure

Appendix A contains the listings of the complete directory structure and the files contained in each directory for the prototype system.

2. Database Generation

The definitions of the databases described in this section are provided in Chapter VI.

a. User Database

The User database consists of two separate hidden files. The Password file is implemented as an Apache-defined *.htpasswd* file. The *htaccess password generator* web tool downloaded from KxS Inc. [16] was used to generate the password hashes. Multiple test users with varying group membership were created for the prototype system and are defined in the *.htpasswd* file. The Group file is implemented as *.groups* and contains only the groups that require access control. The user groups defined for the prototype system are: Administrator, Collaborator, Customer, Developer, Evaluator, and NIST/NSA. The Public group is not specified in the Group file and all users belong to the Public group. The listing of the *.groups* and the *.htpasswd* file can be found in Appendix A.

b. Releasable Items List Database

The Releasable Items List database consists of one file called *ril.txt*. For this implementation the sample set of releasable items includes XML documents, source code, tar archives, and publications. Only one document descriptor file is used for the prototype system; thus for each RIL entry, the same DDF Filename (*ddf_CI0001.xml)* is specified. The listing of the *ril.txt* file can be found in Appendix A.

c. Dissemination Policy Database

The Dissemination Policy database is contained in the file *policy.txt*. The content of this file defines the different combinations of File Type and File Sensitivity markings used by the prototype system to control access to project material. The dissemination policy enforced by the prototype system is summarized in Table 16. The complete listing of the *policy.txt* file can be found in Appendix A.

File Type	File Sensitivity	Authorized User Groups
Code, Engineering Notes, or User Documents	Engineering Release	All user groups except Customer and Public
Verification Evidence	Engineering Release	Only Administrator, Developer, Evaluator, and NIST/NSA

All File Types	Proprietary Restricted	Only Administrator and Developer
All File Types	No Sensitivity Marking (denoted as "*")	All user groups
Publications	Engineering Release	Not released to any groups (denoted as "invalid")

Table 16. Prototype Dissemination Policy

d. Key and Certificate Database

The creation of the Dissemination System public/private key pair and certificate signing request was done two different ways. One method generated a raw key and the other generated an encrypted key. The OpenSSL tool was utilized via the *CA.pl* Perl script to generate all keys and signing requests. The *CA.pl* script was pre-installed with the operating system and provided a user-friendly command line interface for the OpenSSL tool.

Since the CISR CA server does not exist, the *CA.pl* Perl script was also used on the prototype system to simulate the certificate generation and signing functions to be performed by the CISR CA. A self-signed certificate for the CISR CA was first created. The public/private key pair and a certificate signing request were generated next. The OpenSSL tool uses the information in the signing request to generate and sign the Dissemination System certificate. The keys and certificates were created in a directory under the root user home directory. The Dissemination System private key and certificate then replaced the default web server private key and certificate in the Apache defined key and certificate directories. Screen captures of the CA certificate and the Dissemination System certificate are contained in Appendix B.

e. Dissemination Material Repository Database

The Dissemination Material Repository is a distributed database in multiple folders. To create the Dissemination Material Repository multiple test files were imported into the system. These test files were XML documents, non-XML material (e.g. publications and code), HTML files, and document descriptor files. Since the Dissemination Application is not yet implemented, HTML generated views of XML

files were manually generated and imported. The XML and non-XML documents are the files that are disseminated online. The access control markings for the web accessible documents of the prototype system are listed in Table 17. These markings were used in the document descriptor file (*ddf_CI0001.xml*) which is located in Appendix A.

File Name	File Type	File Sensitivity Marking
05paper_tcx.pdf	Publications	* (no sensitivity marking)
DIE-XXE-docbook_2005-04-25.zip	Code	Proprietary Restricted
Documentation_Standards.xml	Engineering Notes	Engineering Release
thesis-jclark.pdf	Publications	Proprietary Restricted
xweb-tangle.py.xweb	Code	Engineering Release
xhtml-review.xsl	Code	Proprietary Restricted

Table 17. Initial Implementation Test Documents

For all the test files, the XML access control tags were located in the same document descriptor file. Only a subset of the possible tag combinations was used for the initial implementation.

The following steps were done manually to simulate the Dissemination Application function that creates redacted XML files and their corresponding HTML files. A folder named *Documentation_Standards_xml* was created in the *html_generated_views* folder of the repository. Then a set of redacted XML files was created in the new folder, one file per authorized user group. The authorized user group name was appended to the filename of the redacted XML file for that particular group. According to the dissemination policy described in Table 16 and the access control tags in the test document descriptor file all non-public groups were allowed access to the test XML file. The per-group redacted XML files that were created are as follows:

- Documentation_Standards_administrator.xml
- Documentation_Standards_collaborator.xml

- Documentation_Standards_customer.xml
- Documentation_Standards_developer.xml
- Documentation_Standards_evaluator.xml
- Documentation_Standards_nist_nsa.xml

An HTML file was also created for each redacted XML file with the same file name except for the file extension. The HTML file contains the XML contents formatted with XSL transformations and an HTML link to the associated redacted XML file. The following HTML files were created.

- Documentation_Standards_administrator.html
- Documentation_Standards_collaborator.html
- Documentation_Standards_customer.html
- Documentation_Standards_developer.html
- Documentation_Standards_evaluator.html
- Documentation_Standards_nist_nsa.html

A screen capture of the *Documentation_Standards_nist_nsa.html* document can be found in Appendix B. The HTML link to the redacted XML file is shown at the end of the HTML view.

f. Webpage Repository Database

The Webpage Repository database consists of all web accessible content. The Dissemination System homepage (*index.html*) contains HTML links to the different group sections of the website. Inside each group section is a group homepage (another *index.html* file) and a set of symbolic links to the following files in the Dissemination Material Repository: redacted XML files and their corresponding HTML files, and original non-XML target files.

The group homepages contain HTML links to the symbolic links in the directory. The creation of the symbolic links is to be performed by the Webpage Manager component of the Dissemination Application, but for the prototype system they

were created manually. Based on the dissemination policy and the access control tags for the non-XML test documents in the document descriptor file, symbolic links pointing to the original non-XML target file were created manually in each group folder. Symbolic links were also created for every redacted XML and HTML file in the *html_generated_views* folder in the appropriate group folder of the Webpage Repository.

For instance in the *collaborator* folder in the Webpage Repository database the following files were created:

- index.html
- Documentation_Standards_collaborator.xml
- Documentation_Standards_collaborator.html
- 05paper_tcx.pdf
- xweb-tangle.py.xweb

All of the files created there are symbolic links except *index.html.* A complete listing of the files for each group directory can be found in Appendix A in the directory structure listing.

g. Audit Log Repository Database

The Apache Web Server has built in logging capability. For the prototype system, the default configuration settings for the Apache audit function were not changed. The default configuration consisted of the following log files:

- access_log
- error_log
- ssl_access_log
- ssl_error_log
- ssl_ request_log

The default configuration for Apache is “LogLevel warn”. This logs all emergency, alert, critical, error, and warning conditions [6]. Different log formats are used by the Apache Web Server as the default log format including the Common Log

Format and the Combined Log Format. The log path specified in the configuration file was */var/httpd/log/*.

3. Manual Simulation of the Dissemination Application

The Dissemination Application program will be implemented as future work. For the prototype all functionality of the Dissemination Application was manually implemented and the processes used to accomplish this task are described in the following sections.

a. Sweeping Handler

The Sweeping Handler is intended to transfer all non-releasable items in the Dissemination Material Repository database into a separate directory for analysis by the system administrator. This function was simulated by comparing the files in the *XML_original* folder and the *non-XML_original* folders with the *ril.txt file*. Any files that were not specified in *ril.txt* were moved into the *revoked* folder. The *linkcheck.pl* Perl script was also run to find broken symbolic links. The script was run from the command line and used the *–a* flag to find all links in the Dissemination System Root folder. Broken links were then repaired or deleted.

b. Redaction Handler

The Redaction Handler is intended to redact the original XML files and generate the proper HTML views of the redacted XML documents. The manual implementation of this function was discussed previously in the Dissemination Material Repository database generation section.

c. Webpage Manager

The Webpage Manager is intended to create symbolic links to files in the Dissemination Material Repository database and updates the group homepages to reflect the current content. The manual implementation of this process was discussed in the Webpage Repository database generation section.

d. Audit Handler

The Audit Handler functions were not implemented in the prototype system.

4. Apache Web Server Configuration

The Apache Web Server came preloaded within the Fedora Core 3 operating system. Additionally, the mod_ssl add-on module was already installed and running on the web server, so no setup of the SSL configuration was necessary to allow HTTPS connections. Since the system could already operate as a standard web server, it only needed to be configured to host the TCX dissemination website. To do so, modifications were made to the Apache configuration file, which can be found in Appendix A. The website root folder was changed to the TCX Webpage Repository root, */var/www/html/tcx/*.

Custom directory definitions were added into the Apache configuration file to specify the Webpage Repository directories that required authentication and access control. Apache supports two directory-based user authentication methods, Basic and Digest. Both methods allow the use of a user name and password for authentication, however, the Digest method is more secure because it only sends password information to the clients in encrypted form. Since the Digest method is not supported by older browsers, most web servers use the Basic method to be compatible with a wide range of browsers. The prototype system also uses the Basic method since the threat of sending a password in the clear to clients is mitigated by the use of the SSL. For the prototype, Basic Authentication was used in conjunction with group-based access control. The group-based access control mechanism allows specific groups to access directories for which they are authorized.

Additionally, some of the Webpage Repository directories needed to be forced to require SSL connections. To do this, a number of Apache RewriteRule directives were needed and implemented. The Apache rewrite engine provides support for runtime URL manipulation and is used by the prototype system to configure specific directories to always require an SSL connection.

5. Administrative and Supporting Tools

a. crond

For the initial implementation the default *crond* configuration file was not modified. The default configuration runs the *logrotate* and *webalizer* daemons on a regular schedule as specified by the configuration files for those daemons.

b. logrotate

For the initial implementation the pre-install *logrotate* configuration file was not modified. The default rotation frequency was set to once a week and the number of archived logs to be maintained was set to four.

c. webalizer

webalizer runs as a daily *cron* job on the Dissemination System. It reads the Apache access log and creates HTML output files with graphs and statistics of use. For the initial implementation, only the Apache access log is processed by *webalizer*. Since the Apache access log only contains audit data for HTTP requests, the *webalizer* output does not include a report on HTTPS requests. Because of this limitation, the review and analysis of audit data for HTTPS requests contained in the Apache SSL logs must be performed manually.

d. OpenSSL

The OpenSSL tool is used to generate the private/public key pair and the certificate signing request as discussed previously in the Key and Certificate database generation sections. The normal mode of operation is to generate a signing request and send the request to the certificate authority for signing. For the prototype system, the Dissemination System acted as the certificate authority and the signing was performed locally.

e. linkcheck.pl

The *linkcheck.pl* Perl script is run manually during the simulation of the Sweeping Handler. The manual running of this tool does not provide the audit logging functionality as defined in the design specification. The *linkcheck.pl* script was obtained from http://www.orlandokuntao.com/mf_linkcheck.html [17]. Its source can be found in Appendix A.

C. TESTING

Three types of functional tests are required for the Dissemination System: Web Server Testing, Dissemination Application Testing, and Tool Testing. The Dissemination Application Testing was not performed.

1. Web Server Functional Testing

Three test scenarios were conducted to verify the following functions of the web server

- User authentication
- Group-based access control
- HTTPS connections for non-public web content

Both the Mozilla web browser running on the Linux Dissemination System and the Microsoft Internet Explorer browser running on a Windows laptop computer were used for testing.

User authentication was verified by logging in as different users and requesting access to both public and non-public web content. All non-public content required authentication with a username and password before the information is served. The login prompt appeared when attempts to access non-public content were made. Access was granted to authorized users who logged in with a correct username and password combination. Negative tests were also conducted to verify that access is denied to unauthorized users by entering incorrect usernames and passwords.

Group-based access control was tested by setting up multiple users with different group memberships. Based on the user's group membership, the user was only authorized to view material cleared for that particular group in accordance with the dissemination policy. Screen captures of the developer group and collaborator group's available content can be found in Appendix B.

HTTPS connections were tested by requesting a normal HTTP connection to non-public web pages. It was observed that the web address in the location bar of the web browser was automatically changed from "http://..." to "https://...". Additionally, user manipulation of the browser's address location to circumvent SSL protection was also tested to verify that once an HTTPS connection is established, the user cannot reload the same protected webpage without using HTTPS. These two tests confirmed that the Apache rewrite directive worked as expected.

2. Tool Testing

The *logrotate*, *webalizer*, and *linkcheck.pl* tools were tested locally on the Linux Dissemination System.

logrotate was verified to be operating correctly through continual use of the system. Audit data was colleted over multiple weeks. After each week, all five audit logs on the system were rotated. The logs were renamed with "_x" appended to their name where "x" ranges from 1 to 4 as defined by the *logrotate* configuration file. New empty audit logs were then created. Daily and hourly rotations were also tested.

The *webalizer* default setting was to update the output HTML files daily. At the end of each month, a month long HTML log analysis file was created. Testing involved viewing the HTML output files and assuring that the contents reflected the Apache access log. A screen capture from the *webalizer* output can be found in Appendix B.

The *linkcheck.pl* script was downloaded and tested prior to incorporating it into the manual simulation of the Dissemination Application. A test directory structure with symbolic links and target files was created. The script was run and found no broken links. Then a target file was removed and the script was run again. The script reported that the symbolic link pointing to the removed target file was broken.

D. PROBLEMS ENCOUNTERED

Two problems involving the limitations on file and terminal device access imposed by the default SELinux policy were encountered. SELinux is automatically enabled in the default installation of Fedora Core 3 Linux. SELinux provides additional security functions to protect the underlying operating system if the Apache server is compromised.

The default SELinx policy disables daemons from communicating with the controlling console [18]. Since Apache's private key is encrypted, the passphrase must be entered when Apache is started at the controlling console. To start Apache, the SELinux policy had to be modified to allow Apache to access the console. After Apache was successfully started, the SELinux policy was reverted to the default policy. The modified SELinux policy file is included in Appendix A.

SELinux also enforces additional access controls on the file system. SELinux will not allow web users to access files without the correct security attribute defined for Apache. Apache will not be able to host files that it does not have authorization to access. This was encountered when files generated locally on the machine but outside of

the Dissemination System Root directory were manually copied into the Dissemination System Root directory. The copied file did not have the correct security attributes for Apache to access. After the attribute was properly set, Apache was able to host the contents.

E. SUMMARY

The development and testing environment was first presented for the prototype system. The construction of the prototype was then described including the manual simulation of the Dissemination Application. Test scenarios, unimplemented functional design features, and challenges encountered were discussed. The next chapter contains the conclusions from this research and future work.

THIS PAGE INTENTIONALLY LEFT BLANK

VII. FUTURE WORK AND CONCLUSION

A. INTRODUCTION

This chapter introduces potential future work and presents the final conclusions on the research and development of the TCX Dissemination System. The design specification presented here is for the initial implementation of the Dissemination System, therefore there is potentially room for improvement. Several features discussed in the requirements analysis (Chapters III and IV) were not implemented and are discussed below along with additional suggestions for improving the design.

B. FUTURE WORK

As previously discussed, a manual simulation of the Dissemination Application was implemented in the Dissemination System initial implementation. Future work should include a complete implementation of the Dissemination Application as specified in the functional design specification. This includes the creation of a Dissemination Application program that fully implements the four primary modules: Sweeping Handler, Redaction Handler, Webpage Manager, and Audit Handler. A programmatic Dissemination Application will allow automatic dissemination of project material once it is imported onto the Dissemination System.

In addition to creating the Dissemination Application program, the development of the operational Dissemination Environment should also be completed (detailed in Chapter III) to fulfill the project goal of open dissemination. While some parts of the environment have been developed for the TCX project, a number of security-relevant mechanisms such as the Releasing Agent, CISR Certificate Authority, and TCX user registration still need to be implemented. With the existence of the fully operational Dissemination Environment, open and seamless online dissemination of TCX project material can easily be accomplished.

While subsequent implementations of the Dissemination Application will include additional audit capabilities, the audit analysis functionality in the initial implementation is incomplete. The initial implementation utilizes the *webalizer* analysis tool in the default configuration that does not process the SSL audit logs created by Apache. The

audit logs contain both valid accesses and illegal attempts to access public and non-public material. However, to provide complete analysis of the logs, follow-on work would need to be developed to customize the *webalizer* configuration to support detailed audit analysis of HTTPS accesses.

To assure the correct design and implementation of the Dissemination System, the next iteration of development should reassess the initial set of requirements and implement a development environment that applies the security assurance requirements as described in Chapter IV. These assurance requirements were not addressed by the functional design specification and initial implementation that was completed for this thesis. As part of the implementation of the assurance requirements, thorough functional testing and vulnerability analysis of the system should be conducted. This would result in a more robust Dissemination System.

The initial implementation of the Dissemination System did not utilize the functionality and benefits of XML to the extent that it could have. Future work on the Dissemination System should use XML to enforce finer grained document control. The granularity for document release of the initial implementation was at the document level; the expanded use of XML in future implementations could afford paragraph-level releases of documents to different user groups. Marking XML documents for paragraph-level release would not require much additional effort, however, the Redaction Handler of the Dissemination Application would require enhancement. Future fine grained document control would require the Redaction Handler to be capable of generating multiple different redacted views for each user group. The redacted views would be from the same source XML document but could potentially contain different sets of paragraphs from that document.

Although a discussion of document and file signatures was included in Chapter III as a dissemination objective of the TCX project, this initial implementation of the Dissemination System used test documents that did not contain integrity signatures. Future work should include the implementation of signed dissemination material and the external distribution of the signature verification tool. For finer grained document release, digital signatures should be implemented on the smallest releasable portion of a

file. This will result in a single file containing multiple signatures within it, but the use of XML rather than other word processed documents will make the implementation more practicable.

C. CONCLUSION

After analyzing existing web-based dissemination systems for open source projects it was determined that the TCX Dissemination System should be developed from the ground up to achieve the open dissemination requirements of the TCX project. The concept of operation for an XML-centric web-based dissemination environment was created to map all interactions and data flow among different components that make up the Dissemination System. Development of the requirements was preceded by an analysis of the assumptions, threats, policies, and security objectives for the Dissemination System. This development framework was based on the Common Criteria methodology, but was not rigorously followed because of the non-evaluatability of the system. The requirements included both security functional and security assurance requirements for the Dissemination System. These requirements were addressed in the functional design specification for the initial implementation. The Dissemination System consists of three classes of system processes running on a Linux-based operating system: web server, Dissemination Application, and administrative tools. The web server implements group-based access control, user authentication, and secure communication. The Dissemination Application is a program that automates the redaction and preparation of releasable materials for dissemination. Future designs will utilize this design specification as a backbone. The Dissemination System design specification is the blueprint for the initial implementation prototype.

A useable dissemination prototype was created that satisfies a subset of the TCX dissemination requirements. The problems encountered, and their solutions, provided insight into what future developers might encounter when configuring a dissemination system on a Linux-based machine. The prototype also generated future work for subsequent developments.

The next design iteration should include a complete implementation of the Dissemination Environment which will allow seamless dissemination of TCX project material. Additional future work will ensure that all dissemination requirements for the

TCX project are achieved and an example of high assurance development will be available worldwide to geographically distributed developers.

APPENDIX A – FILE LISTINGS OF PROTOTYPE SYSTEM

A. Introduction

This appendix contains the directory structure listing and the following files in order:

- *.groups*
- *.htpasswd*
- *ril.txt*
- *policy.txt*
- *ddf_CI0001.xml*
- *httpd.conf*
- *linkcheck.pl*
- *apache.te* (SELinux Apache policy)

Changes made to the files presented here relative to those of the existing prototype system are for formatting reasons only. Portions of *httpd.conf* and *apache.te* contain no changes and were removed for readability purposes. These two files are marked accordingly.

B. Directory Structure

This directory structure includes all directories containing database material relevant to the Dissemination System. Additionally it contains directories and files associated with the configuration of the system. Lines ending with a ":" denote that the lines following it are contained within that directory. Lines ending with a "/" denote a directory. Lines ending with a "@" denote a symbolic link.

```
/var/www/:
./
../
documents/
html/
login/
usage/

/var/www/documents:
./
../
```

```
ddf/
ds_label.tar
html_generated_views/
non-XML_original/
policy/
revoked/
ril/
XML_original/

/var/www/documents/ddf:
./
../
ddf_CI0001.xml

/var/www/documents/html_generated_views:
./
../
Documentation_Standards_xml/

/var/www/documents/html_generated_views/Documentation_Standards_xml:
./
../
Documentation_Standards_administrator.html
Documentation_Standards_administrator.xml
Documentation_Standards_collaborator.html
Documentation_Standards_collaborator.xml
Documentation_Standards_developer.html
Documentation_Standards_developer.xml
Documentation_Standards_evaluator.html
Documentation_Standards_evaluator.xml
Documentation_Standards_nist_nsa.html
Documentation_Standards_nist_nsa.xml

/var/www/documents/non-XML_original:
./
../
05paper_tcx.pdf
DIE-XXE-docbook_2005-04-25.zip
thesis-jclark.pdf
xhtml-review.xsl
xweb-tangle.py.xweb

/var/www/documents/policy:
./
../
policy.txt

/var/www/documents/revoked:
./
../

/var/www/documents/ril:
./
../
ril.txt

/var/www/documents/XML_original:
```

```
./
../
Documentation_Standards.xml

/var/www/html:
./
../
tcx/

/var/www/html/tcx:
./
../
administrator/
collaborator/
customer/
developer/
evaluator/
favicon.ico
index.html
nist_nsa/
public/

/var/www/html/tcx/administrator:
./
../
05paper_tcx.pdf@
DIE-XXE-docbook_2005-04-25.zip@
Documentation_Standards_administrator.html@
Documentation_Standards_administrator.xml@
index.html
thesis-jclark.pdf@
xhtml-review.xsl@
xweb-tangle.py.xweb@

/var/www/html/tcx/collaborator:
./
../
05paper_tcx.pdf@
Documentation_Standards_collaborator.html@
Documentation_Standards_collaborator.xml@
index.html
xweb-tangle.py.xweb@

/var/www/html/tcx/customer:
./
../
05paper_tcx.pdf@
index.html

/var/www/html/tcx/developer:
./
../
05paper_tcx.pdf@
DIE-XXE-docbook_2005-04-25.zip@
Documentation_Standards_developer.html@
Documentation_Standards_developer.xml@
index.html
```

```
thesis-jclark.pdf@
xhtml-review.xsl@
xweb-tangle.py.xweb@

/var/www/html/tcx/evaluator:
./
../
05paper_tcx.pdf@
Documentation_Standards_evaluator.html@
Documentation_Standards_evaluator.xml@
index.html
xweb-tangle.py.xweb@

/var/www/html/tcx/nist_nsa:
./
../
05paper_tcx.pdf@
Documentation_Standards_nist_nsa.html@
Documentation_Standards_nist_nsa.xml@
index.html
xweb-tangle.py.xweb@

/var/www/html/tcx/public:
./
../
05paper_tcx.pdf@
index.html

/var/www/login:
./
../
.groups
.htpasswd

/var/www/usage:
./
../
ctry_usage_200504.png
ctry_usage_200505.png
ctry_usage_200506.png
daily_usage_200504.png
daily_usage_200505.png
daily_usage_200506.png
hourly_usage_200504.png
hourly_usage_200505.png
hourly_usage_200506.png
index.html
msfree.png
usage_200504.html
usage_200505.html
usage_200505_May12_1401.html
usage_200505.save.html
usage_200506.html
usage.png
webalizer.png

/var/log/httpd:
```

```
./
../
access_log
access_log.1
access_log.2
access_log.3
access_log.4
error_log
error_log.1
error_log.2
error_log.3
error_log.4
ssl_access_log
ssl_access_log.1
ssl_access_log.2
ssl_access_log.3
ssl_access_log.4
ssl_error_log
ssl_error_log.1
ssl_error_log.2
ssl_error_log.3
ssl_error_log.4
ssl_request_log
ssl_request_log.1
ssl_request_log.2
ssl_request_log.3
ssl_request_log.4

/etc/httpd/:
./
../
build@
conf/
conf.d/
logs@
modules@
run@

/etc/httpd/conf:
./
../
httpd.conf
httpd.conf.orig
httpd.conf.tcx
magic
Makefile@
ssl.crt/
ssl.key/

/etc/httpd/conf/ssl.crt:
./
../
Makefile.crt
server.crt
server.crt.ds
server.crt.orig
```

```
/etc/httpd/conf/ssl.key:
./
../
server.key
server.key.ds
server.key.orig

/etc/httpd/conf.d:
./
../
auth_kerb.conf
auth_mysql.conf
auth_pgsql.conf
authz_ldap.conf
htdig.conf
mailman.conf
manual.conf
mrtg.conf
perl.conf
php.conf
python.conf
README
squirrelmail.conf
ssl.conf
subversion.conf
webalizer.conf
welcome.conf
wordtrans.conf
```

C. .groups

The following listing shows the content of the *.groups* file which is part of the User database.

```
# ************************************************************************
# TCX Project Dissemination System Prototype
# .groups - created by D. Rob Kane on 2005 05 25
# This file contains the group names and members of the groups
# ************************************************************************

administrator: drkane irvine
collaborator: drkane irvine
customer: drkane tdnguyen
developer: drkane tdnguyen
evaluator: drkane jclark
nist_nsa: drkane jclark
```

D. .htpasswd

The following listing shows the content of the *.htpasswd* file, which is part of the User database.

```
# ************************************************************************
```

```
# TCX Project Dissemination System Prototype
# .htpasswd - created by D. Rob Kane on 2005 05 25
# Password hashes were created using KxS Inc. htaccess password
# generator.
# http://www.kxs.net/support/htaccess_pw.html
# ***************************************************************************

drkane:0MnLIhv/6aVPQ
irvine:LAqPOSf4F61xE
jclark:GgBiPMEGc4/qs
tdnguyen:dPwBXJWqG6NgQ
```

E. ril.txt

The following listing shows the content of the *ril.txt* file, which is the Releasable Items List database.

```
# ***************************************************************************
# TCX Project Dissemination System Prototype
# ril_v1.0 - created by D. Rob Kane on 2005 05 25
# The following file is the Releasable Items List Database used specify
# which files are available for dissemination.
# ***************************************************************************

/non-XML_original/DIE-XXE-docbook_2005-04-25.zip      ddf_CI0001.xml
/XML_original/Documentation_Standards.xml             ddf_CI0001.xml
/non-XML_original/thesis-jclark.pdf                   ddf_CI0001.xml
/non-XML_original/xhtml-review.xsl                    ddf_CI0001.xml
/non-XML_original/xweb-tangle.py.xweb                 ddf_CI0001.xml
/non-XML_original/05paper_tcx.pdf                     ddf_CI0001.xml
```

F. policy.txt

The following listing shows the content of the *policy.txt* file which is the Dissemination Policy database. This listing was changed for formatting purposes.

```
# ***************************************************************************
# TCX Project Dissemination System Prototype
# policy.txt - created by D. Rob Kane on 2005 05 25
# The following file is the policy database.  The first column lists
# all possible access tags available.  The second column contains the
# groups that have access to those tags.
# ***************************************************************************

CODE ENGINEERING RELEASE                  administrator, collaborator,
                                          developer, evaluator,
                                          nist_nsa
ENGINEERING NOTES ENGINEERING RELEASE     administrator, collaborator,
                                          developer, evaluator,
                                          nist_nsa
PUBLICATIONS ENGINEERING RELEASE          invalid
SPECIFICATIONS ENGINEERING RELEASE        administrator, collaborator,
                                          developer, evaluator,
                                          nist_nsa
```

```
USER DOCUMENTS ENGINEERING RELEASE        administrator, collaborator,
                                          developer, evaluator,
                                          nist_nsa
VERIFICATION EVIDENCE ENGINEERING RELEASE administrator, developer,
                                          evaluator, nist_nsa

CODE PROPRIETARY RESTRICTED               administrator, developer
ENGINEERING NOTES PROPRIETARY RESTRICTED  administrator, developer
PUBLICATIONS PROPRIETARY RESTRICTED       administrator, developer
SPECIFICATIONS PROPRIETARY RESTRICTED     administrator, developer
USER DOCUMENTS PROPRIETARY RESTRICTED     administrator, developer
VERIFICATION EVIDENCE PROPRIETARY RESTRICTED    administrator,
                                                developer

CODE *                  administrator, collaborator, customer,
                        developer, evaluator, nist_nsa, public
ENGINEERING NOTES *     administrator, collaborator, customer,
                        developer, evaluator, nist_nsa, public
PUBLICATIONS *          administrator, collaborator, customer,
                        developer, evaluator, nist_nsa, public
SPECIFICATIONS *        administrator, collaborator, customer,
                        developer, evaluator, nist_nsa, public
USER DOCUMENTS *        administrator, collaborator, customer,
                        developer, evaluator, nist_nsa, public
VERIFICATION EVIDENCE * administrator, collaborator, customer,
                        developer, evaluator, nist_nsa, public
```

G. ddf_CI0001.xml

The following listing shows the content of the *ddf_CI0001.xml* file, which is the document descriptor file for the initial implementation of the Dissemination System.

```
<controls xmlns=
      "tag:tcx.cisr.nps.navy.mil,2005-05-02:dissemination_labeling"
      xmlns:xlink="http://www.w3.org/1999/xlink">
  <resource xlink:href="DIE-XXE-docbook_2005-04-25.zip">
    <label>CODE</label>

    <label>PROPRIETARY RESTRICTED</label>
  </resource>

  <resource xlink:href="Documentation_Standards.html">
    <label>ENGINEERING NOTES</label>

    <label>ENGINEERING RELEASE</label>
  </resource>

  <resource xlink:href="Documentation_Standards.xml">
    <label>ENGINEERING NOTES</label>

    <label>ENGINEERING RELEASE</label>
  </resource>

  <resource xlink:href="thesis-jclark.pdf">
    <label>PUBLICATIONS</label>
```

```
      <label>PROPRIETARY RESTRICTED</label>
    </resource>

    <resource xlink:href="xweb-tangle.py.xweb">
      <label>CODE</label>

      <label>ENGINEERING RELEASE</label>
    </resource>

    <resource xlink:href="xhtml-review.xsl">
      <label>CODE</label>

      <label>PROPRIETARY RESTRICTED</label>
    </resource>

    <resource xlink:href="05paper_tcx.pdf">
      <label>PUBLICATIONS</label>

      <label></label>
    </resource>

  </controls>
```

H. *httpd.conf*

The following listing shows the content of the *httpd.conf* file, which is the Apache configuration file. The headers at the beginning of the file contain comments describing the changes made to the document. Comment blocks were added throughout the file to document TCX-specific modifications. Portions of this document were not changed and were removed for readability.

```
#
# Based upon the NCSA server configuration files originally by Rob McCool.
#
# This is the main Apache server configuration file.  It contains the
# configuration directives that give the server its instructions.
# See <URL:http://httpd.apache.org/docs-2.0/> for detailed information about
# the directives.
#
# Do NOT simply read the instructions in here without understanding
# what they do.  They're here only as hints or reminders.  If you are unsure
# consult the online docs. You have been warned.
#
# The configuration directives are grouped into three basic sections:
#  1. Directives that control the operation of the Apache server process as a
#     whole (the 'global environment').
#  2. Directives that define the parameters of the 'main' or 'default' server,
#     which responds to requests that aren't handled by a virtual host.
#     These directives also provide default values for the settings
#     of all virtual hosts.
#  3. Settings for virtual hosts, which allow Web requests to be sent to
#     different IP addresses or hostnames and have them handled by the
```

```
#     same Apache server process.
#
# Configuration and logfile names: If the filenames you specify for many
# of the server's control files begin with "/" (or "drive:/" for Win32), the
# server will use that explicit path.  If the filenames do *not* begin
# with "/", the value of ServerRoot is prepended -- so "logs/foo.log"
# with ServerRoot set to "/etc/httpd" will be interpreted by the
# server as "/etc/httpd/logs/foo.log".
#
# ********************************************************************
# TCX Project Dissemination System Prototype
# Created by D. Rob Kane on 2005 05 25
# Modifications:
#  1.  Rewrite rules for SSL in DocumentRoot directory
#  2.  Changed DocumentRoot
#  3.  Setup access controls for directories within DocumentRoot
# ********************************************************************

### Section 1: Global Environment
#
# The directives in this section affect the overall operation of Apache,
# such as the number of concurrent requests it can handle or where it
# can find its configuration files.
#

.
.
.
NO CHANGES – REMOVED FOR READABILITY
.
.
.

### Section 2: 'Main' server configuration
#
# The directives in this section set up the values used by the 'main'
# server, which responds to any requests that aren't handled by a
# <VirtualHost> definition.  These values also provide defaults for
# any <VirtualHost> containers you may define later in the file.

.
.
.
NO CHANGES – REMOVED FOR READABILITY
.
.
.

# DocumentRoot: The directory out of which you will serve your
# documents. By default, all requests are taken from this directory, but
# symbolic links and aliases may be used to point to other locations.
# ********************************************************************
# TCX Project Dissemination System Prototype
# The DocumentRoot was changed to the TCX folder
# ********************************************************************
#
DocumentRoot "/var/www/html/tcx/"
```

```
# ************************************************************************
# TCX Project Dissemination System Prototype
# The following rewrite rules assure that https is used for the six
# directories listed below.
# ************************************************************************
RewriteEngine on
RewriteCond    %{SERVER_PORT} !^443$
RewriteRule ^/administrator(.*)$ https://localhost/administrator$1 [L,R]
RewriteRule ^/collaborator(.*)$ https://localhost/collaborator$1 [L,R]
RewriteRule ^/customer(.*)$ https://localhost/customer$1 [L,R]
RewriteRule ^/developer(.*)$ https://localhost/developer$1 [L,R]
RewriteRule ^/evaluator(.*)$ https://localhost/evaluator$1 [L,R]
RewriteRule ^/nist_nsa(.*)$ https://localhost/nist_nsa$1 [L,R]

#
# Each directory to which Apache has access can be configured with respect
# to which services and features are allowed and/or disabled in that
# directory (and its subdirectories).
#
# First, we configure the "default" to be a very restrictive set of
# features.
#
<Directory />
    Options FollowSymLinks
    AllowOverride All
</Directory>

#
# Note that from this point forward you must specifically allow
# particular features to be enabled - so if something's not working as
# you might expect, make sure that you have specifically enabled it
# below.
#

# ************************************************************************
# TCX Project Dissemination System Prototype
# The DocumentRoot was changed to the TCX folder
# ************************************************************************
# This should be changed to whatever you set DocumentRoot to.
#
<Directory "/var/www/html/tcx/">

#
# Possible values for the Options directive are "None", "All",
# or any combination of:
#   Indexes Includes FollowSymLinks SymLinksifOwnerMatch ExecCGI MultiViews
#
# Note that "MultiViews" must be named *explicitly* --- "Options All"
# doesn't give it to you.
#
# The Options directive is both complicated and important.  Please see
# http://httpd.apache.org/docs-2.0/mod/core.html#options
# for more information.
#
    Options Indexes FollowSymLinks
```

```
#
# AllowOverride controls what directives may be placed in .htaccess files.
# It can be "All", "None", or any combination of the keywords:
#   Options FileInfo AuthConfig Limit
#
    AllowOverride All

#
# Controls who can get stuff from this server.
#
    Order allow,deny
    Allow from all

</Directory>

# ************************************************************************
# TCX Project Dissemination System Prototype
# The following sections specify the access control based on group affiliation.
# This uses the Apache basic authentication with group based control.
# ************************************************************************
<Directory "/var/www/html/tcx/administrator">
  AuthUserFile /var/www/login/.htpasswd
  AuthGroupFile /var/www/login/.groups
  AuthName "Administrator"
  AuthType Basic
  require group administrator
</Directory>

<Directory "/var/www/html/tcx/collaborator">
  AuthUserFile /var/www/login/.htpasswd
  AuthGroupFile /var/www/login/.groups
  AuthName "Collaborator"
  AuthType Basic
  require group collaborator
</Directory>

<Directory "/var/www/html/tcx/customer">
  AuthUserFile /var/www/login/.htpasswd
  AuthGroupFile /var/www/login/.groups
  AuthName "Customer"
  AuthType Basic
  require group customer
</Directory>

<Directory "/var/www/html/tcx/developer">
  AuthUserFile /var/www/login/.htpasswd
  AuthGroupFile /var/www/login/.groups
  AuthName "Developer"
  AuthType Basic
  require group developer
</Directory>

<Directory "/var/www/html/tcx/evaluator">
  AuthUserFile /var/www/login/.htpasswd
  AuthGroupFile /var/www/login/.groups
  AuthName "Evaluator"
  AuthType Basic
```

```
  require group evaluator
</Directory>

<Directory "/var/www/html/tcx/nist_nsa">
  AuthUserFile /var/www/login/.htpasswd
  AuthGroupFile /var/www/login/.groups
  AuthName "NIST/NSA"
  AuthType Basic
  require group nist_nsa
</Directory>

#
# UserDir: The name of the directory that is appended onto a user's home
# directory if a ~user request is received.
#
# The path to the end user account 'public_html' directory must be
# accessible to the webserver userid.  This usually means that ~userid
# must have permissions of 711, ~userid/public_html must have permissions
# of 755, and documents contained therein must be world-readable.
# Otherwise, the client will only receive a "403 Forbidden" message.
#

.
.
.
NO FURTHER CHANGES – REMOVED FOR READABILITY
.
.
.
```

I. *linkcheck.pl*

linkcheck.pl is the Perl script used by the Dissemination System to remove broken symbolic links. Because it was downloaded from the Internet, the complete script is contained in this appendix.

```
#!/usr/bin/perl -w
#
# SccsId[] = "@(#)linkcheck.pl 1.3 01/08/03 (Link check Perl program)"
#
#-----------------------------------------------------------------------#
#                            linkcheck.pl                               #
# ---------------------------------------------------------------------- #
#                                                                        #
#   Copyright (c) 2002-2003 by Bob Orlando.  All rights reserved.        #
#                                                                        #
#   Permission to use, copy, modify and distribute this software         #
#   and its documentation for any purpose and without fee is hereby      #
#   granted, provided that the above copyright notice appear in all      #
#   copies, and that both the copyright notice and this permission       #
#   notice appear in supporting documentation, and that the name of      #
#   Bob Orlando not be used in advertising or publicity pertaining       #
#   to distribution of the software without specific, written prior      #
#   permission.  Bob Orlando makes no representations about the          #
```

```
#   suitability of this software for any purpose.  It is provided      #
#   "as is" without express or implied warranty.                       #
#                                                                      #
#   BOB ORLANDO DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS           #
#   SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY      #
#   AND FITNESS.  IN NO EVENT SHALL BOB ORLANDO BE LIABLE FOR ANY      #
#   SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES          #
#   WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER    #
#   IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION,     #
#   ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF     #
#   THIS SOFTWARE.                                                     #
#                                                                      #
# -------------------------------------------------------------------- #
#        Program documentation and notes located at the bottom.        #
#--------------------------------------------------------------------#

   BEGIN { $diagnostics::PRETTY = 1 }

   $SIG{'INT'}=sub {print "\nExiting on $SIG{'INT'}\n";exit $SIG{'INT'}};

   use File::Find;
   use Getopt::Std;
   use Cwd;
   use POSIX qw(uname);
   my $host =  (uname)[1];
   $| = 1; # Autoflush (unbuffer output)

   use vars qw($opt_a $opt_H $opt_h $opt_l $opt_r $opt_v);
   my $options='aHhlrv';
   exit_usage("Invalid option!\n") unless (getopts($options));
   show_documentation() if ($opt_H); # Full documentation
   exit_usage()         if ($opt_h); # or usage brief.
   exit_usage("Filesystem required.\n") if ($#ARGV < 0);

   if ($opt_v)
   {
      use diagnostics;
   }

 #-----------------------------------------------------------#
 # Eliminate all but local filesystem searches right away. #
 #-----------------------------------------------------------#
   my $local_fs;
   my @search;
   foreach (@ARGV)
   {
      if ($local_fs = `df -lk $_`)
      {
         push(@search, $_);
      }
      else
      {
         print "File system $_ must be local to $host, not NFS mounted.",
               "\nSkipping $_.\n";
         $_ = "";
      }
   }
```

```
#-------------------------------------------------------------#
# Ignore find command's stderr output (eliminates "Permission #
# denied" and most--not ALL--other bothersome messages).      #
#-------------------------------------------------------------#
  open(OLDERR, ">&STDERR");
  open(STDERR, ">/dev/null") or die "Can't redirect stderr: $!";

  my $q = 0; # Found counter
  my $r = 0; # Removed counter
  find sub # [Anonymous] subroutine reference (called a coderef).
  {
     return unless -l "$_"; # Skip all but links.

   #-----------------------------------------------------------#
   # Skip nfs mounted links, and /proc and /cdrom pathnames.  #
   #-----------------------------------------------------------#
     return if (
                 (lstat("$_"))[0] < 0
               ||
                 $File::Find::name =~ /\/proc/s
               ||
                 $File::Find::name =~ /\/cdrom/s
               );

   #-----------------------------------------------------------#
   # Skip link if it's not on a local filesystem as well.     #
   #-----------------------------------------------------------#
     my $dir = cwd;
     return unless ($local_fs = `df -lk $dir`);

     $! = 0; # Clear error message variable
     return unless defined(my $target = readlink("$_"));

     my $error  = "$!";
        $error  = "($error)" if (defined($error) && $error ne "");

     my $ls_out = ($opt_l)
       ? `ls -albd $File::Find::name 2> /dev/null`
       : "$File::Find::name -> $target";

     chomp($ls_out);

     unless (-e "$target") # Unless the link is OK, do the following.
     {
        $q++;
        print "Broken link: $ls_out $error\n";
        if ($opt_r)
        {
           print  "rm '$File::Find::name'\n";
           if (unlink("$File::Find::name") == 0) # Zero = none deleted.
           {
              print "Unable to remove $File::Find::name!\n";
              return;
           }
           $r++;
           print "Removed '$File::Find::name'\n" if ($opt_v);
```

```
        }
        return;
      }

    #-------------------------------------------------------------#
    # Return unless user requests list of all links (-a).      #
    #-------------------------------------------------------------#
      return unless ($opt_a);

      if    (-f "$target") { print "Linked file: $ls_out $error\n"; }
      elsif (-d "$target") { print "Linked dir:  $ls_out $error\n"; }
      elsif (-l "$target") { print "Linked link: $ls_out $error\n"; }
      elsif (-p "$target") { print "Linked pipe: $ls_out $error\n"; }
      elsif (-S "$target") { print "Linked sock: $ls_out $error\n"; }
      elsif (-b "$target") { print "Linked dev:  $ls_out $error\n"; }
      elsif (-c "$target") { print "Linked char: $ls_out $error\n"; }
      elsif (-t "$target") { print "Linked tty:  $ls_out $error\n"; }
      else                 { print "Linked ???:  $ls_out $error\n"; }

      $error = "";
      return;
   }, @search; # find sub

 #---------------------------------------------------------------#
 # Restore stderr.                                               #
 #---------------------------------------------------------------#
   close(STDERR) or die "Can't close STDERR: $!";
   open( STDERR, ">&OLDERR") or die "Can't restore stderr: $!";
   close(OLDERR) or die "Can't close OLDERR: $!";

   print "$host: Found $q broken links.  Removed $r.\n";
   exit 1;


#=====================================================================#
#             S U B R O U T I N E S  /  F U N C T I O N S             #
#                       (in alphabetical order)                       #
#---------------------------------------------------------------------#
sub exit_usage # Exits with non-zero status.                          #
               # Global vars:   $main::notify                         #
               #                $main::support                        #
#---------------------------------------------------------------------#
{
   my $fn_name = "exit_usage";
   my $txt     ;

 #---------------------------------------------------------------#
 # Assign to private variable, $notify either $main::support or  #
 # $main::notify (takes $main::support over $main::notify).      #
 #---------------------------------------------------------------#
   my $notify;
   if (defined($ENV{LOGNAME} )) { $notify = $ENV{LOGNAME}; }
   else                         { $notify = $ENV{USER};    }

   $txt =  "Usage:   $0 -$options fs ...\n";
   $txt =  "$_[0]\n$txt" if ($#_ >= 0); # Prefix message arguments
   $txt .= "\n         -a = Display All links."
```

```
        .  "\n         -H = Displays full documentation."
        .  "\n         -h = Gives usage brief."
        .  "\n         -l = Long list (e.g. 'ls -al')."
        .  "\n         -r = Remove broken links (use with caution)."
        .  "\n         -v = Verbose output."
        .  "\n         fs = Required filesystem for search."
        .  "\n              (multiple filesystems may be specified)\n"
        .  "\nPurpose: Search filesystem (descending directories) for"
        .  "\n         broken links, optionally displaying all links"
        .  "\n         (-a) and/or removing (-r) them.\n";

 #-------------------------------------------------------------------#
 # If running interactively, then give'm usage, else notify      #
 # program support person(s) because a cron'd job called usage.  #
 #-------------------------------------------------------------------#
   print "$txt";

   exit 1;
} # sub exit_usage

#-------------------------------------------------------------------------#
sub show_documentation # Display program documentation at bottom.      #
#-------------------------------------------------------------------------#
{
   my $n = 0;
   foreach (my @doc_lines = <main::DATA>)
   {
      print "$_";
   }
   exit $n;
} # sub show_documentation


__END__ # Documentation section follows:
#=========================================================================#
#                       D O C U M E N T A T I O N                         #
#=========================================================================#
#                                                                         #
#      Author: Bob Orlando                                                #
#                                                                         #
#        Date: April 29, 2002                                             #
#                                                                         #
#  Program ID: linkcheck.pl                                               #
#                                                                         #
#     Purpose: Search local filesystem or systems                         #
#              (descending directories) for broken                        #
#              links, optionally displaying all links                     #
#              (-a) and/or removing (-r) them.                            #
#                                                                         #
#       Usage: linkcheck.pl -aHhlrv fs ...                                #
#                           -a = Display All links.                       #
#                           -H = Detailed documentation.                  #
#                           -h = Usage brief.                             #
#                           -l = Long list (e.g. 'ls -al').               #
#                           -r = Remove broken links                      #
#                                (use with caution).                      #
#                           -v = Verbose output.                          #
```

```
#                           fs = Required filesystem for search        #
#                                (multiple filesystems may be          #
#                                specified)                            #
#                                                                      #
#    Examples: linkcheck.pl /        # Lists broken links (short list) #
#              linkcheck.pl -l /     # Lists broken links (long list)  #
#              linkcheck.pl -a /home # Lists all links in /home.       #
#              linkcheck.pl -r /usr  # Removes broken links from /usr. #
#                                                                      #
#     Returns: Zero on success.                                        #
#              Nonzero in failure.                                     #
#                                                                      #
#       Files: ................................................        #
#              ................................................        #
#                                                                      #
#       Notes: ................................................        #
#              ................................................        #
#                                                                      #
#    Modified: 2003-01-08 Bob Orlando                                  #
#                v1.3   * Force autoflush (unbuffer output).           #
#                                                                      #
#              2002-06-14 Bob Orlando                                  #
#                v1.2   * Add tally of links found and removed.        #
#                       * Add host name to messages.                   #
#                       * Initialize counters $q and $r to 0;          #
#                                                                      #
#              2002-06-04 Bob Orlando                                  #
#                v1.1   * Initial SCCS release.                        #
#                                                                      #
#----------------------------------------------------------------------#
```

J. *apache.te*

The *apache.te* file contains the security policy for SELinux as modified for the TCX Dissemination System. The headers at the beginning of the file contain comments describing the changes made to the document. Portions of this document were not changed and were removed for readability.

```
#
# X-Debian-Packages: apache2-common apache
#
##############################################################################
#
# Policy file for running the Apache web server
#
# NOTES:
#  This policy will work with SUEXEC enabled as part of the Apache
#  configuration. However, the user CGI scripts will run under the
#  system_u:system_r:httpd_$1_script_t domain where $1 is the domain of the
#  of the creating user.
#
#  The user CGI scripts must be labeled with the httpd_$1_script_exec_t
#  type, and the directory containing the scripts should also be labeled
```

```
#  with these types. This policy allows user_r role to perform that
#  relabeling. If it is desired that only sysadm_r should be able to relabel
#  the user CGI scripts, then relabel rule for user_r should be removed.
#
################################################################################
# ******************************************************************************
# TCX Project Dissemination System Prototype
# apache.te - modified by D. Rob Kane on 2005 05 25
# Modifications:
#  1.  Allow httpd to access tty and pty devices
# ******************************************************************************
type http_port_t, port_type, reserved_port_type;

bool httpd_unified false;

# Allow httpd cgi support
bool httpd_enable_cgi false;

# Allow httpd to read home directories
bool httpd_enable_homedirs false;

# Run SSI execs in system CGI script domain.
bool httpd_ssi_exec false;

.
.
.
NO CHANGES – REMOVED FOR READABILITY
.
.
.

########################################
# When the admin starts the server, the server wants to acess
# the TTY or PTY associated with the session. The httpd appears
# to run correctly without this permission, so the permission
# are dontaudited here.
################################################
dontaudit httpd_t admin_tty_type:chr_file rw_file_perms;

# ******************************************************************************
# TCX Project Dissemination System Prototype
# Allow httpd to access tty and pty devices
# ******************************************************************************
allow httpd_t { admin_tty_type }:chr_file { getattr ioctl read write };

allow httpd_t krb5_conf_t:file { getattr read };
dontaudit httpd_t krb5_conf_t:file { write };

.
.
.
NO FURTHER CHANGES – REMOVED FOR READABILITY
.
.
.
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B -- SCREEN CAPTURES OF PROTOTYPE SYSTEM

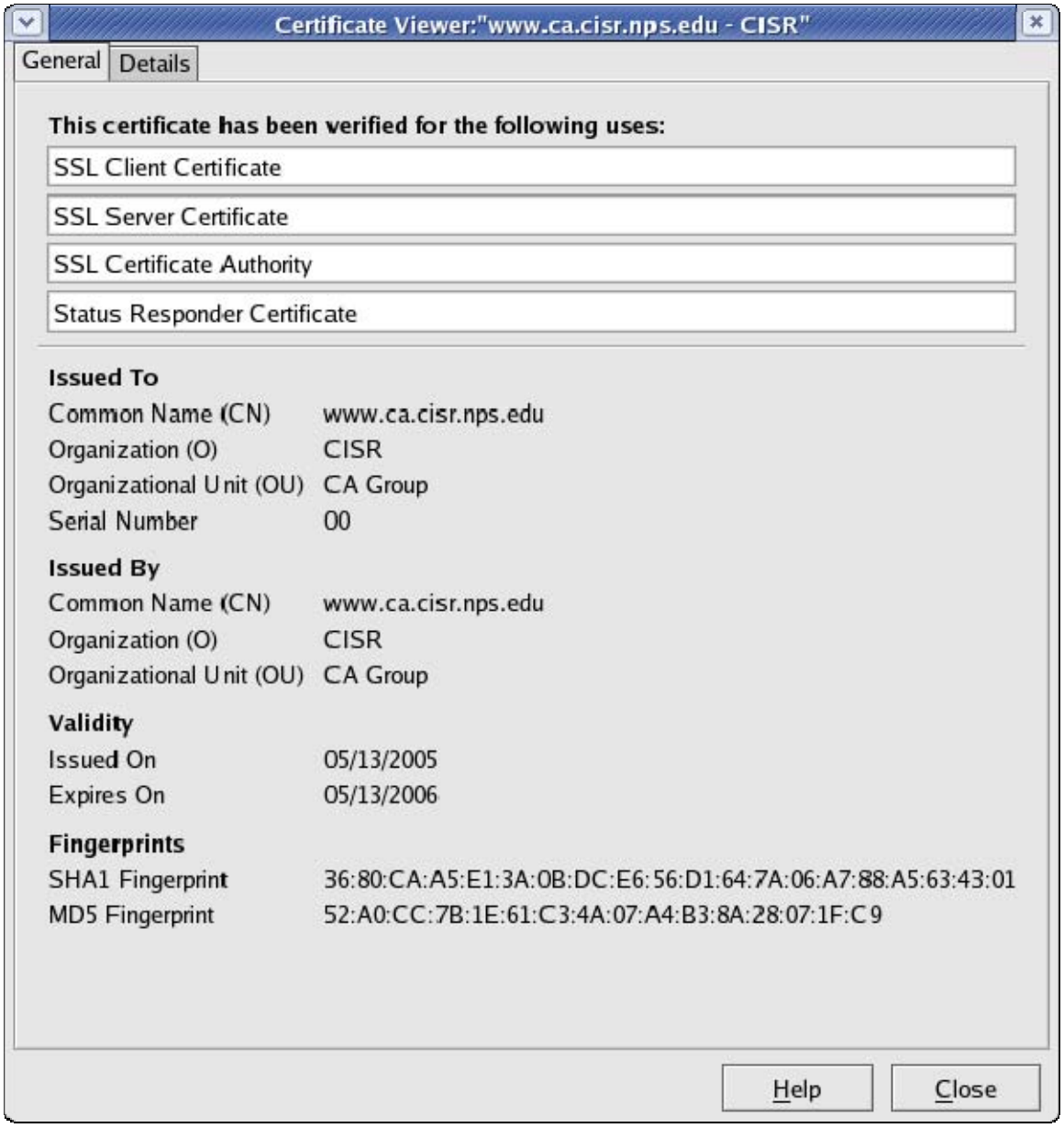
A. Introduction

The following screen captures are included in this appendix:

- CISR CA certificate
- Dissemination System certificate
- Documentation_Standards.html
- Developer web content
- Collaborator web content
- *webalizer* output

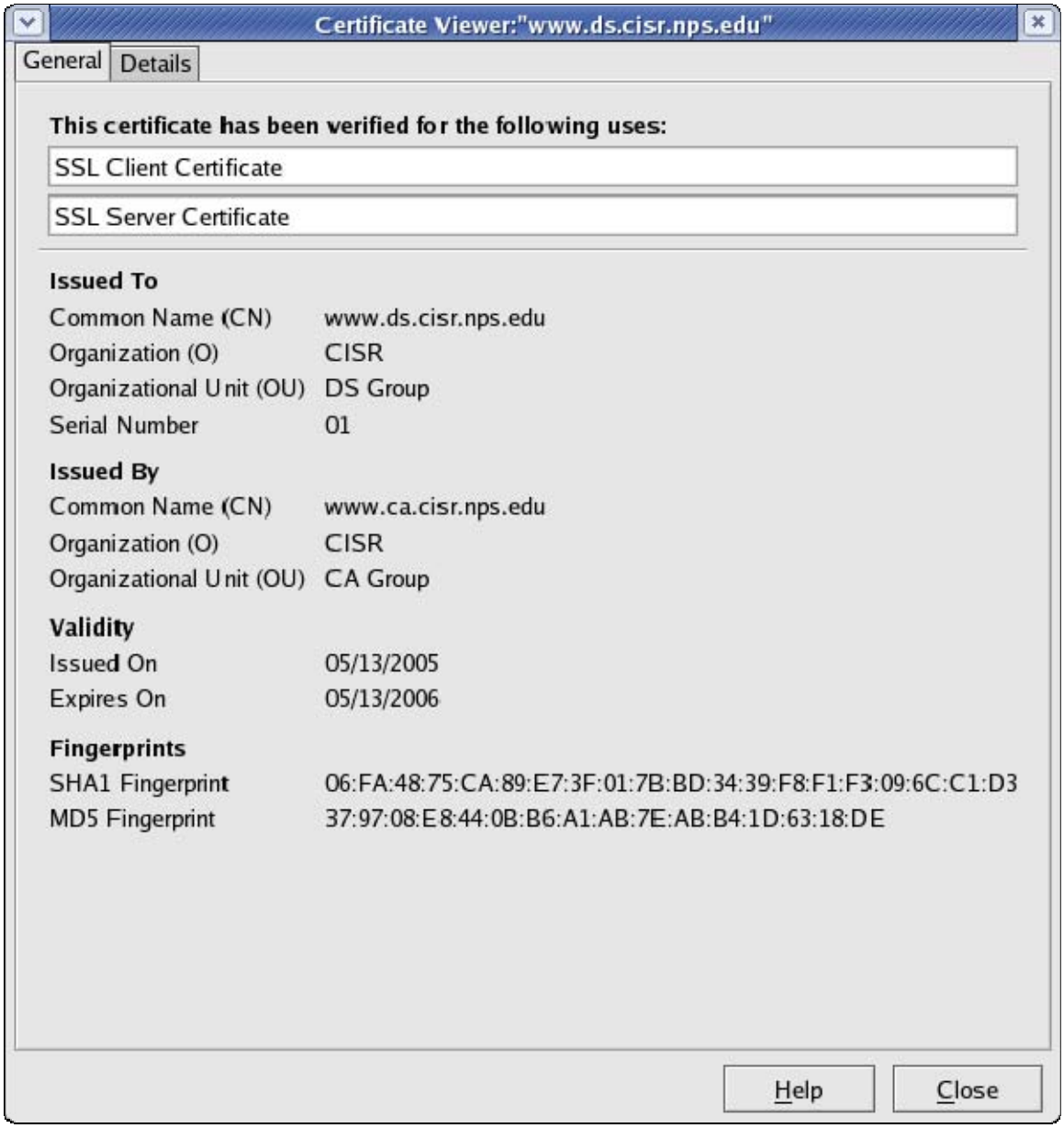
B. CISR CA Certificate

The following screen capture is the test CISR CA digital certificate created for the prototype system:



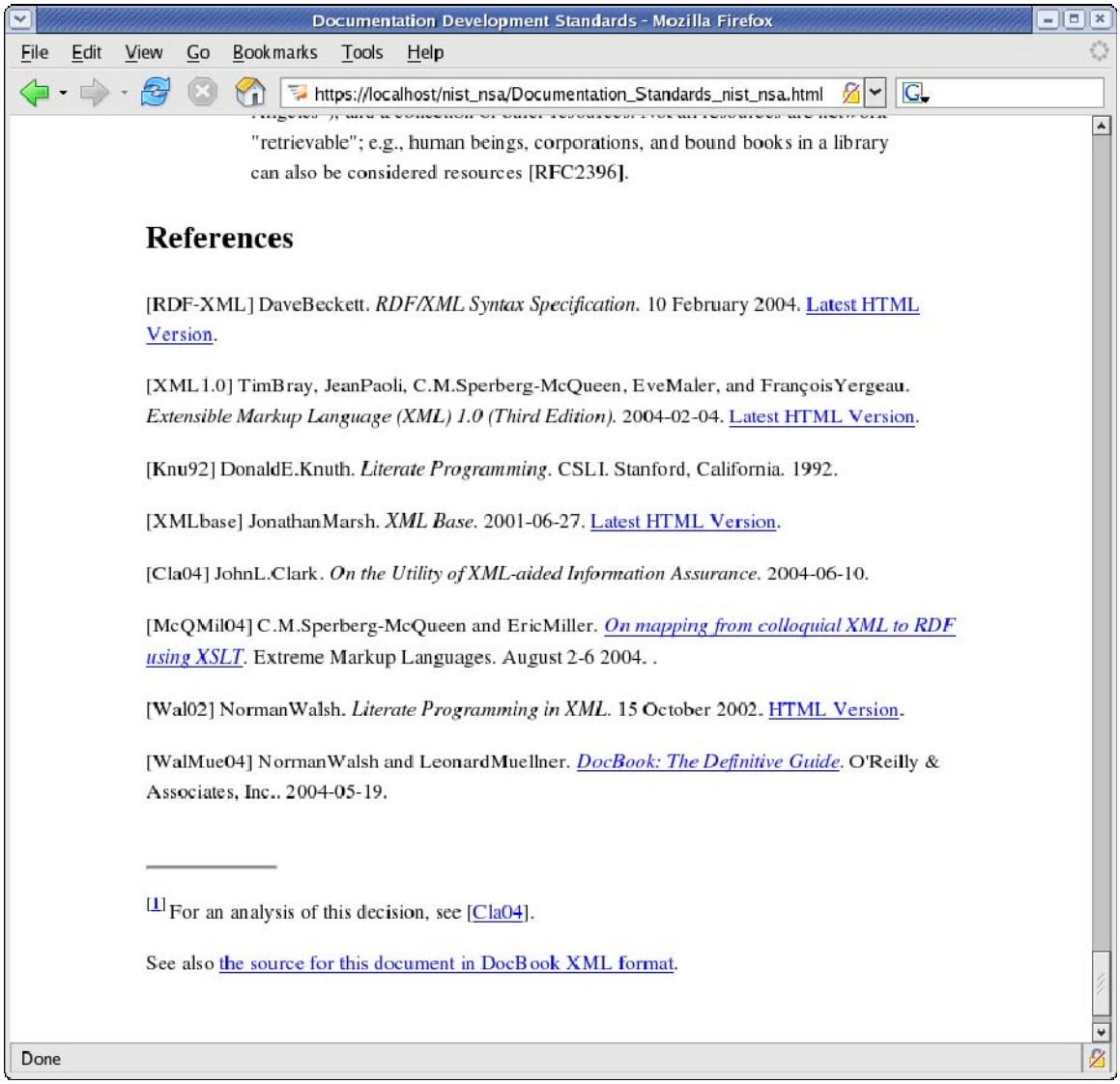
C. Dissemination System Certificate

The following screen capture is the test Dissemination System digital certificate created for the prototype system:



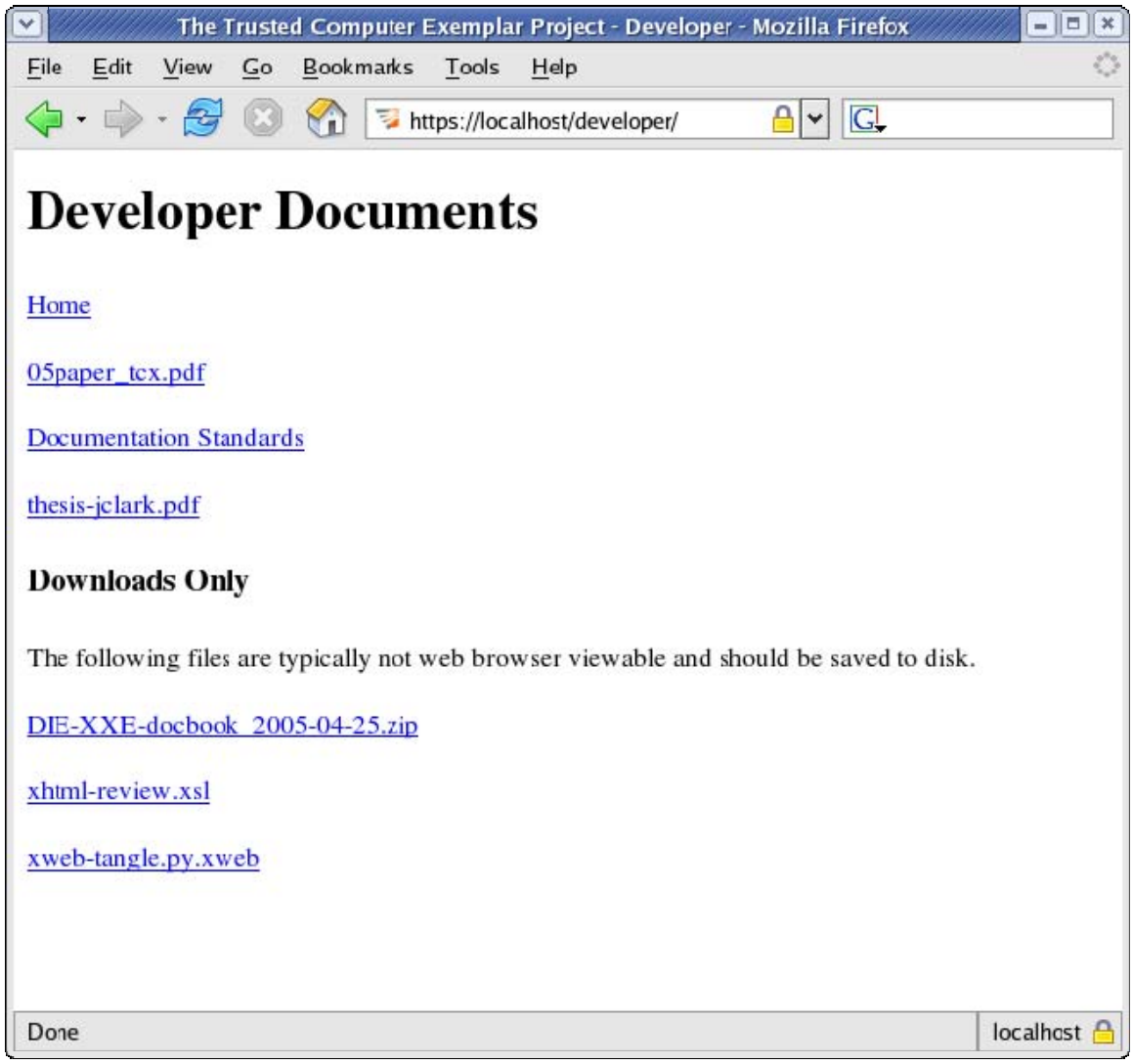
D. Documentation_Standards_nist_nsa.html

The following screen capture shows an excerpt of a test HTML generated view that contains the HTML link to the source XML document for signature verification purposes:



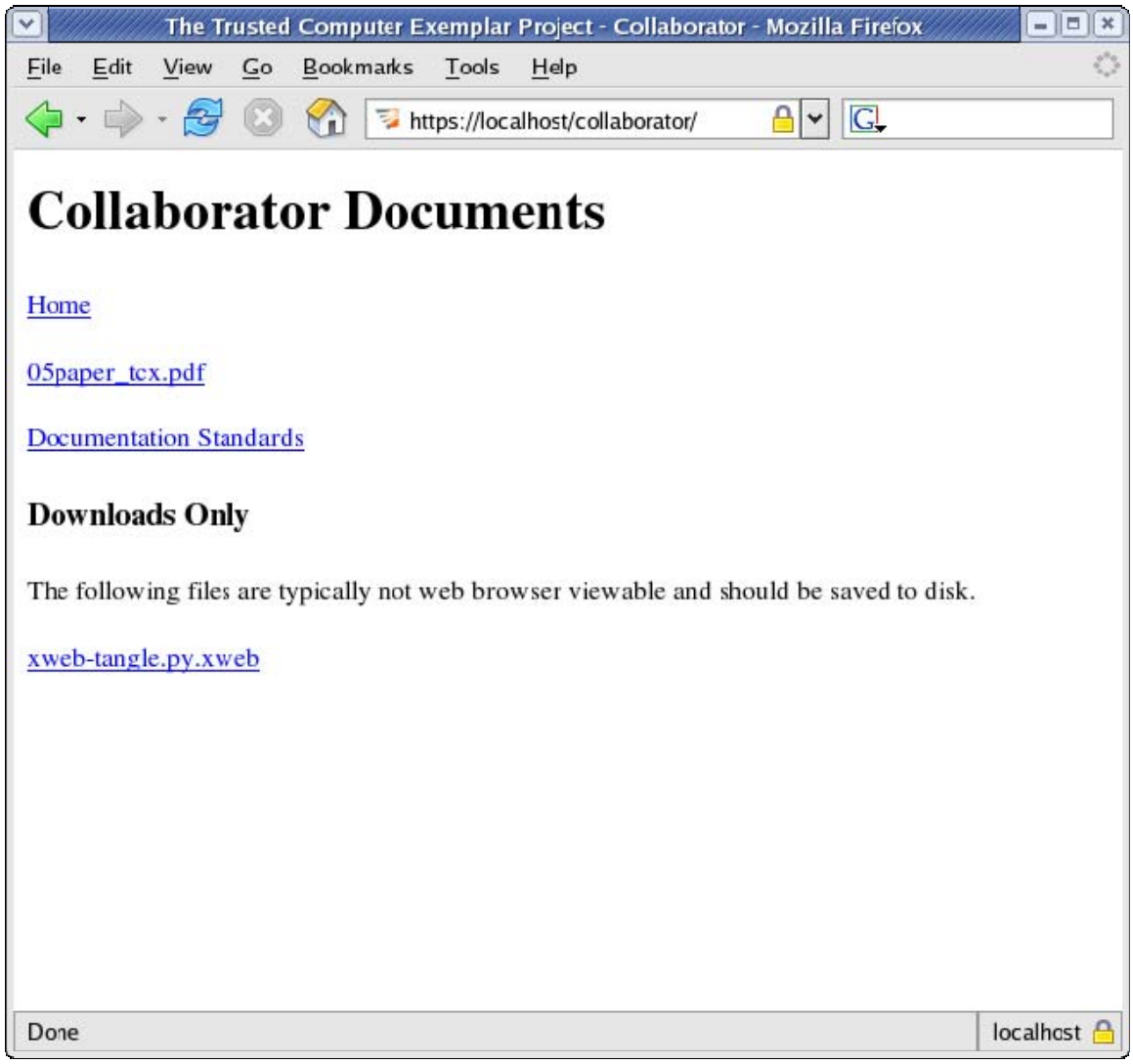
E. Developer Web Content

The following screen capture shows the test dissemination material available to the developer group on the Dissemination System:



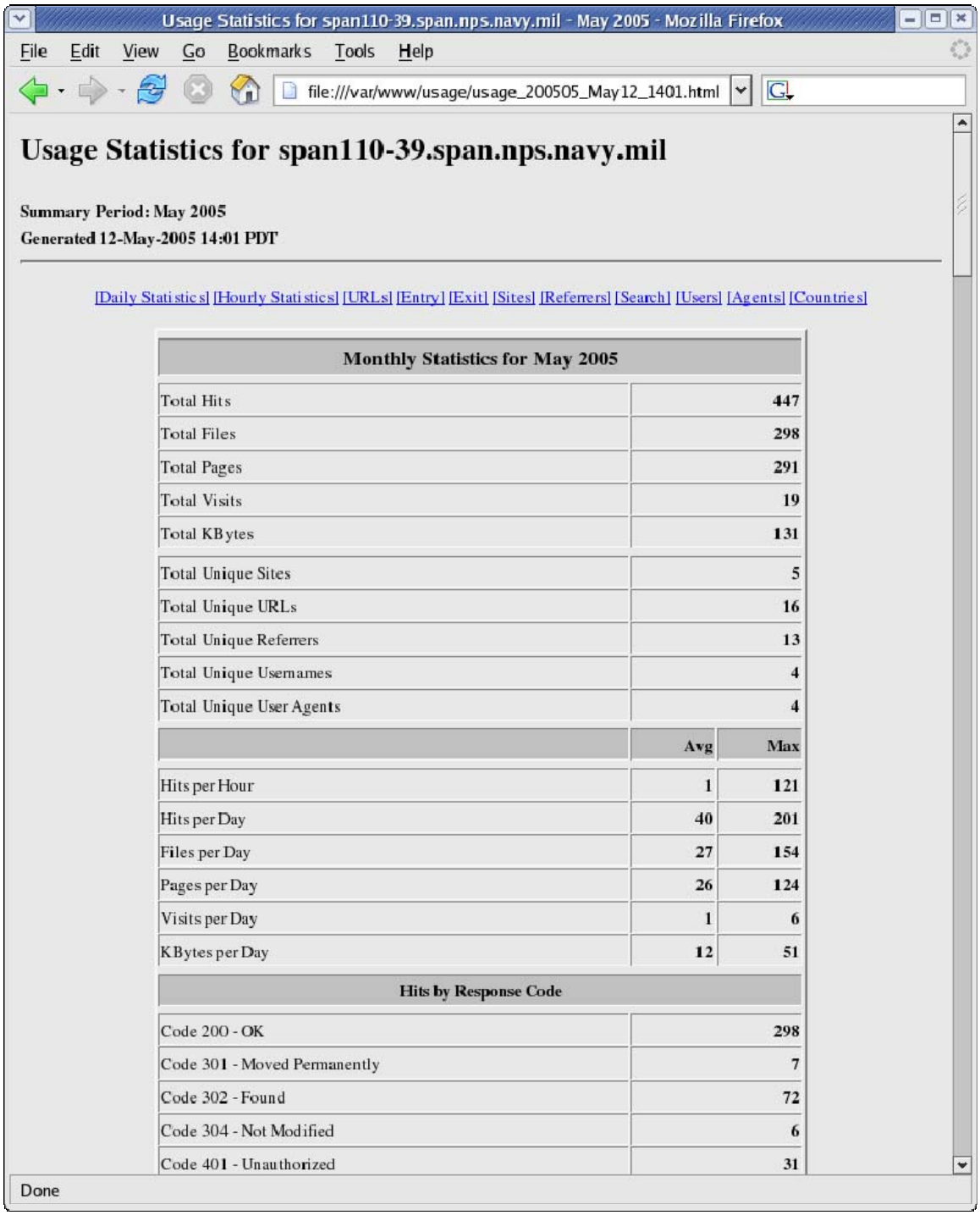
F. Collaborator Web Content

The following screen capture shows the test dissemination material available to the collaborator group on the Dissemination System:



G. *webalizer* Output

The following three screen captures highlight some of the output generated by the *webalizer* tool.

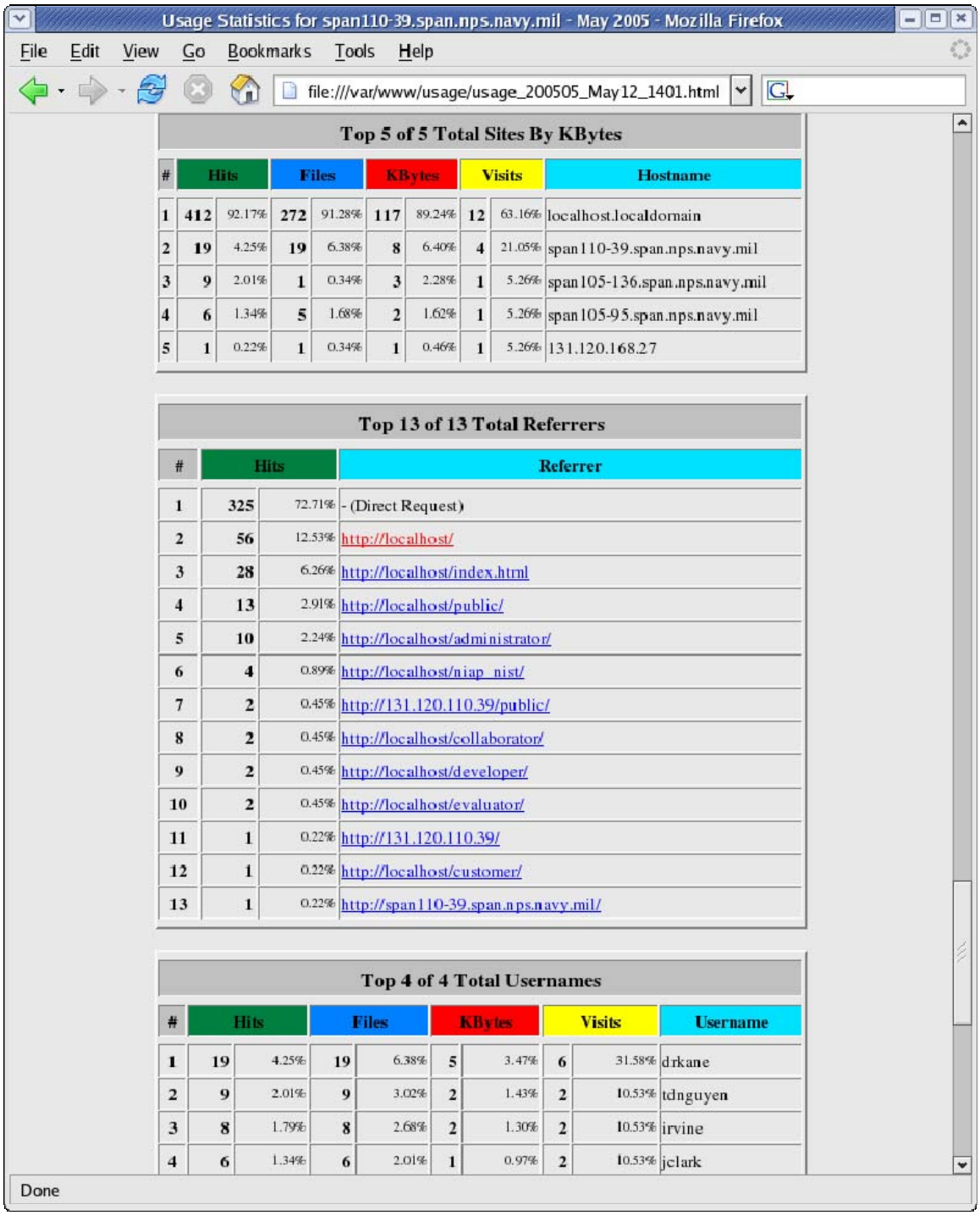


Usage Statistics for span110-39.span.nps.navy.mil

Summary Period: May 2005
Generated 12-May-2005 14:01 PDT

[Daily Statistics] [Hourly Statistics] [URLs] [Entry] [Exit] [Sites] [Referrers] [Search] [Users] [Agents] [Countries]

Monthly Statistics for May 2005		
Total Hits		447
Total Files		298
Total Pages		291
Total Visits		19
Total KBytes		131
Total Unique Sites		5
Total Unique URLs		16
Total Unique Referrers		13
Total Unique Usernames		4
Total Unique User Agents		4
	Avg	Max
Hits per Hour	1	121
Hits per Day	40	201
Files per Day	27	154
Pages per Day	26	124
Visits per Day	1	6
KBytes per Day	12	51
Hits by Response Code		
Code 200 - OK		298
Code 301 - Moved Permanently		7
Code 302 - Found		72
Code 304 - Not Modified		6
Code 401 - Unauthorized		31



Top 5 of 5 Total Sites By KBytes

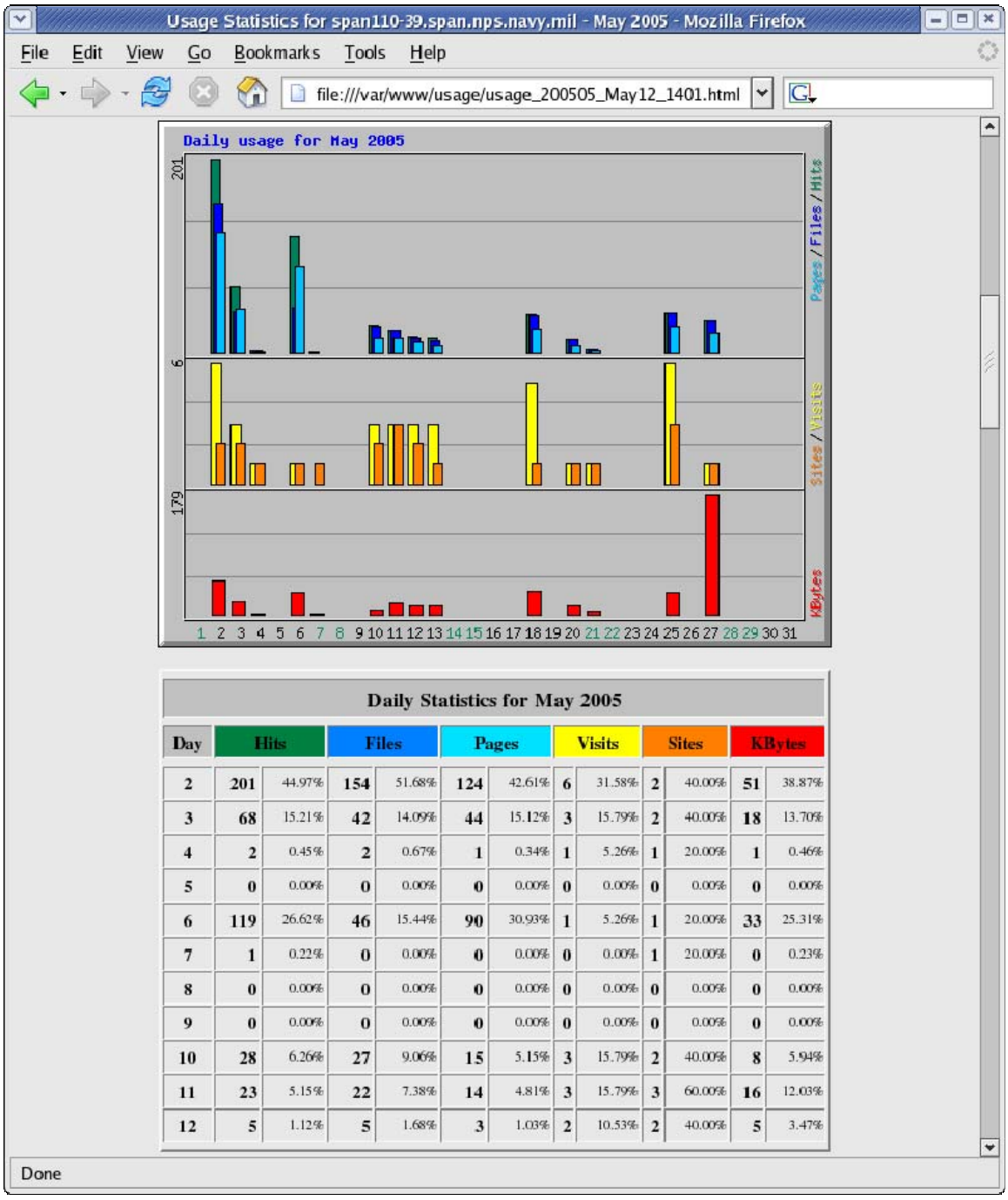
#	Hits		Files		KBytes		Visits		Hostname
1	412	92.17%	272	91.28%	117	89.24%	12	63.16%	localhost.localdomain
2	19	4.25%	19	6.38%	8	6.40%	4	21.05%	span110-39.span.nps.navy.mil
3	9	2.01%	1	0.34%	3	2.28%	1	5.26%	span105-136.span.nps.navy.mil
4	6	1.34%	5	1.68%	2	1.62%	1	5.26%	span105-95.span.nps.navy.mil
5	1	0.22%	1	0.34%	1	0.46%	1	5.26%	131.120.168.27

Top 13 of 13 Total Referrers

#	Hits		Referrer
1	325	72.71%	- (Direct Request)
2	56	12.53%	http://localhost/
3	28	6.26%	http://localhost/index.html
4	13	2.91%	http://localhost/public/
5	10	2.24%	http://localhost/administrator/
6	4	0.89%	http://localhost/niap_nist/
7	2	0.45%	http://131.120.110.39/public/
8	2	0.45%	http://localhost/collaborator/
9	2	0.45%	http://localhost/developer/
10	2	0.45%	http://localhost/evaluator/
11	1	0.22%	http://131.120.110.39/
12	1	0.22%	http://localhost/customer/
13	1	0.22%	http://span110-39.span.nps.navy.mil/

Top 4 of 4 Total Usernames

#	Hits		Files		KBytes		Visits		Username
1	19	4.25%	19	6.38%	5	3.47%	6	31.58%	drkane
2	9	2.01%	9	3.02%	2	1.43%	2	10.53%	tdnguyen
3	8	1.79%	8	2.68%	2	1.30%	2	10.53%	irvine
4	6	1.34%	6	2.01%	1	0.97%	2	10.53%	jclark



Daily Statistics for May 2005

Day	Hits		Files		Pages		Visits		Sites		KBytes	
2	201	44.97%	154	51.68%	124	42.61%	6	31.58%	2	40.00%	51	38.87%
3	68	15.21%	42	14.09%	44	15.12%	3	15.79%	2	40.00%	18	13.70%
4	2	0.45%	2	0.67%	1	0.34%	1	5.26%	1	20.00%	1	0.46%
5	0	0.00%	0	0.00%	0	0.00%	0	0.00%	0	0.00%	0	0.00%
6	119	26.62%	46	15.44%	90	30.93%	1	5.26%	1	20.00%	33	25.31%
7	1	0.22%	0	0.00%	0	0.00%	0	0.00%	1	20.00%	0	0.23%
8	0	0.00%	0	0.00%	0	0.00%	0	0.00%	0	0.00%	0	0.00%
9	0	0.00%	0	0.00%	0	0.00%	0	0.00%	0	0.00%	0	0.00%
10	28	6.26%	27	9.06%	15	5.15%	3	15.79%	2	40.00%	8	5.94%
11	23	5.15%	22	7.38%	14	4.81%	3	15.79%	3	60.00%	16	12.03%
12	5	1.12%	5	1.68%	3	1.03%	2	10.53%	2	40.00%	5	3.47%

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

[1] Irvine, C. E., Levin, T. E., Nguyen, T. D., and Dinolt, G. W., “The Trusted Computing Exemplar Project”, *Proceedings of the 2004 IEEE Systems Man and Cybernetics Information Assurance Workshop*, West Point, NY, June 2004, pp. 109-115.

[2] Common Criteria Project Sponsoring Organizations, "Common Criteria for Information Technology Security Evaluation," CCIMB-2004-01-001, Version 2.2, January 2004. http://www.commoncriteriaportal.org, Accessed: June 2005.

[3] Nguyen, T. D., Levin, T. E., and Irvine, C. E., "TCX Project: High Assurance for Secure Embedded Systems", SIGBED Review, Volume 2, Number 2, April 2005, Special Issue on IEEE RTAS 2005 Work-in-Progress, http://www.cs.virginia.edu/sigbed/vol2_num2.html, Accessed: June 2005.

[4] Shapiro, J., “EROS: The Extremely Reliable Operating System,” 1999, http://www.eros-os.org, Accessed: May 2005.

[5] “The Fiasco Microkernel,” February 2004, http://os.inf.tu-dresden.de/fiasco/, Accessed: May 2005.

[6] “The Apache Software Foundation,” 2005, http://www.apache.org, Accessed: June 2005.

[7] “OpenBSD,” 2005, http://www.openbsd.org, Accessed: May 2005.

[8] Department of Defense. “Department of Defense Trusted Computer System Evaluation Criteria – 5200.28-STD,” 26 December 1985, http://www.radium.ncsc.mil/tpep/library/rainbow/5200.28-STD.pdf, Accessed: May 2005.

[9] National Security Agency, "U.S. Government Protection Profile for Separation Kernels in Environments Requiring High Robustness," Version 0.621, 1 July 2004, http://niap.nist.gov/pp/draft_pps/pp_draft_skpp_hr_v0.621.html, Accessed: June 2005.

[10] Department of Defense. “A Guide to Understanding Trusted Distribution in Trusted Systems – NCSC-TG-008,” 15 December 1988,

http://www.radium.ncsc.mil/tpep/library/rainbow/NCSC-TG-008.pdf, Accessed: May 2005.

[11] Walsh, N. and Muellner. L., “DocBook.org,” 2005, http://www.docbook.org, Accessed: May 2005.

[12] National Security Agency, "U.S. Government Web Server Protection Profile for Basic Robustness Environments," Version 0.41, 1 August 2003, http://niap.nist.gov/pp/draft_pps/pp_draft_websrv_br_v0.41.pdf, Accessed: June 2005.

[13] Vixie, P. “*crond*(8) – Linux man page” 2005, http://www.die.net/doc/linux/man/man8/crond.8.html, Accessed: June 2005.

[14] Troan, E. and Brown, P. “*logrotate*(8) – Linux man page” 2005, http://www.die.net/doc/linux/man/man8/logrotate.8.html, Accessed: June 2005.

[15] Barrett, B. L. “*webalizer*(1) – Linux man page” 2005, http://www.die.net/doc/linux/man/man1/webalizer.1.html, Accessed: June 2005.

[16] KxS Inc. “Htaccess Password Generator,” 2004, http://www.kxs.net/support/htaccess_pw.html, Accessed: June 2005.

[17] Bob Orlando. “Finding Broken Symbolic/Soft Links with linkcheck.pl,” September 4, 2004, http://www.orlandokuntao.com/mf_linkcheck.html, Accessed: June 2005.

[18] “Fedora Core 3 SELinux FAQ,” March 3, 2005, http://fedora.redhat.com/docs/selinux-faq-fc3/index.html, Accessed: June 2005.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, VA

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, CA

3. Hugo A. Badillo
   NSA
   Fort Meade, MD

4. George Bieber
   OSD
   Washington, DC

5. RADM Joseph Burns
   Fort George Meade, MD

6. John Campbell
   National Security Agency
   Fort Meade, MD

7. Deborah Cooper
   DC Associates, LLC
   Roslyn, VA

8. CDR Daniel L. Currie
   PMW 161
   San Diego, CA

9. Louise Davidson
   National Geospatial Agency
   Bethesda, MD

10. Vincent J. DiMaria
    National Security Agency
    Fort Meade, MD

11. LCDR James Downey
    NAVSEA
    Washington, DC

12. Dr. Diana Gant
National Science Foundation
Arlington, VA

13. Jennifer Guild
SPAWAR
Charleston, SC

14. Richard Hale
DISA
Falls Church, VA

15. LCDR Scott D. Heller
SPAWAR
San Diego, CA

16. Wiley Jones
OSD
Washington, DC

17. Russell Jones
N641
Arlington, VA

18. David Ladd
Microsoft Corporation
Redmond, WA

19. Dr. Carl Landwehr
National Science Foundation
Arlington, VA

20. Steve LaFountain
NSA
Fort Meade, MD

21. Dr. Greg Larson
IDA
Alexandria, VA

22. Penny Lehtola
NSA
Fort Meade, MD

23. Ernest Lucier
Federal Aviation Administration

Washington, DC

24. CAPT Deborah A. McGhee
Headquarters U.S. Navy
Arlington, VA

25. Dr. Vic Maconachy
NSA
Fort Meade, MD

26. Doug Maughan
Department of Homeland Security
Washington, DC

27. Dr. John Monastra
Aerospace Corporation
Chantilly, VA

28. John Mildner
SPAWAR
Charleston, SC

29. Jim Roberts
Central Intelligence Agency
Reston, VA

30. Keith Schwalm
Good Harbor Consulting, LLC
Washington, DC

31. Charles Sherupski
Sherassoc
Round Hill, VA

32. Dr. Ralph Wachter
ONR
Arlington, VA

33. David Wirth
N641
Arlington, VA

34. Daniel Wolf
NSA
Fort Meade, MD

35. Jim Yerovi
NRO
Chantilly, VA

36. CAPT Robert Zellmann
CNO Staff N614
Arlington, VA

37. Dr. Cynthia E. Irvine
Naval Postgraduate School
Monterey, CA

38. Thuy D. Nguyen
Naval Postgraduate School
Monterey, CA

39. Douglas R. Kane Jr.
Naval Postgraduate School
Monterey, CA